
UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

---

# Rilevamento di situazioni di pericolo per donne e bambini mediante l'uso di reti neurali convoluzionali (CNN)

***Detection of dangerous situations for women and children using convolutional neural networks (CNN).***

**RELATORI:**
Prof. Aldo Franco Dragoni

**LAUREANDO:**
Mattia Ducci

**CORRELATORI:**
Dott. Paolo Sernani
Dott. Paolo Contardo

---

Anno accademico 2023 - 2024

# Sommario

Negli ultimi anni, il problema della violenza sulle donne e sui bambini ha registrato un aumento preoccupante. Spesso, le vittime non denunciano queste violenze per paura, rendendo difficile l'intervento delle forze dell'ordine. Questo studio propone un metodo che va a rilevare in maniera autonoma situazioni di pericolo per donne e bambini, utilizzando il deep learning (apprendimento profondo), in particolare le reti neurali convoluzionali (CNN).
Il mio approccio prevede l'addestramento delle reti neurali utilizzando rappresentazioni grafiche dei segnali audio, in particolare gli spettrogrammi in scala Mel. Prima di essere trasformati nella rappresentazione grafica, gli audio vengono "puliti" per ridurre o annullare i rumori di fondo, migliorando così la precisione del modello. L'obbiettivo è far sì che il modello impari automaticamente a riconoscere le situazioni di pericolo da quelle non pericolose e, in caso di pericolo, a identificare se le vittime sono donne o bambini. La ricerca prevede di testare diverse tipologie di reti CNN per determinare quale offra risultati più affidabili e accurati per l'implementazione di questa tecnologia.
Questo studio potrebbe fornire uno strumento prezioso per migliorare la sicurezza di questi soggetti, consentendo un intervento tempestivo in situazioni di emergenza. I risultati ottenuti hanno dimostrato che queste reti riescono con buona accuratezza a rilevare situazioni di emergenza, con la speranza che in futuro si possano migliorare ulteriormente i risultati ottenuti da questa ricerca.

# Indice

# Capitolo 1

# Introduzione

La sicurezza delle donne dei bambini è un tema di crescente rilevanza, in considerazione delle persistenti e preoccupanti violenze che questi gruppi subiscono quotidianamente. Le cronache quotidiane e i rapporti ufficiali rivelano una realtà drammatica e allarmante. Secondo il servizio analisi del Dipartimento di Pubblica Sicurezza e Save the Children Italia, nel 2023 le richieste di aiuto per episodi di violenza domestica subita dalle donne sono state 13.793. Dove nel 61,5% dei casi l'autore risulta legato alla vittima da una relazione di tipo sentimentale, attuale o passata. Si registrano inoltre numerosi casi (2.124) di violenza in cui le presunte vittime sono minori, divisi in maniera abbastanza equa tra i due sessi (51.1% femmine e 48.7% maschi). In molti di questi casi, i minori risultano essere testimoni di violenze domestiche perpetrate contro le donne, mentre in altri sono le vittime dirette di abusi.

Questo quadro evidenzia l'urgenza di sviluppare soluzioni innovative ed efficaci per la prevenzione e l'intervento in situazioni di violenza. La crescente consapevolezza della gravità di queste problematiche sottolinea la necessità di approcci tecnologici avanzati, che possano supportare le forze dell'ordine e le organizzazioni di protezione nell'identificazione e gestione di situazioni di rischio. Per cercare di ovviare a questo problema ho deciso di utilizzare il deep learning nel campo della classificazione dei suoni.

La classificazione audio è il processo di analisi e categorizzazione degli audio in varie categorie, che nel nostro caso sono: donne in pericolo, bambini in pericolo e situazioni senza pericolo. Questo approccio sfrutta le reti CNN per riconoscere pattern

e caratteristiche nei dati audio, permettendo di distinguere tra diverse categorie di suoni. Questo tipo di tecnologia può essere applicata su numerosi campi:

- Musica e streaming, permette di riconoscere brani musicali e fornisce suggerimenti su brani simili a quelli ascoltati dall'utente.
- Sanità, per monitorare i suoni respiratori e cardiaci o per il rilevamento di crisi epilettiche in base ai suoni emessi dai pazienti.
- Assistenti vocali, come Google, Alexa e Siri, che riconoscono e rispondono a comandi vocali.
- Trascrizione del parlato, permette di trasformare il parlato in testo, utile per persone con disabilità.
- Conservazione della fauna, per monitorare le specie a rischio o per lo studio degli ecosistemi.
- Sicurezza, permette di rilevare suoni di emergenza come allarmi e urla. Inoltre, può essere utilizzata per monitorare pericoli in ambienti domestici, rilevando situazioni di rischio per donne e bambini, e inviando allarmi alle autorità competenti.

Nonostante la sua potenza il deep learning presenta alcuni svantaggi. Per esempio, richiede enormi quantità di dati per l'addestramento. Questo significa raccogliere un gran numero di segnali audio etichettati e trasformarli in rappresentazioni visive, un compito che è costoso e laborioso. Fortunatamente, esistono dei metodi per aumentare il numero degli audio del dataset con l'uso di tecniche di aumento dei dati.
Il campo di interesse per questa ricerca è quello della sicurezza. Questo studio cerca di proporre una tecnologia che potrebbe risultare utile alle forze dell'ordine per prevenire situazioni di violenza. Il progetto si occupa di identificare situazioni che potrebbero rappresentare un pericolo, semplicemente analizzando un audio contenente delle urla. Vengono testati diversi modelli di reti neurali convoluzionali (CNN), tra cui ResNet50 [1], ResNet101 [1], Xception [2], MobileNetV2 [3] e InceptionV3 [4], noti per la loro efficacia nel riconoscimento delle immagini, che verranno

utilizzati per classificare gli audio. Tali modelli riceveranno in ingresso delle immagini ottenute trasformando gli audio in rappresentazioni tempo-frequenza, come spettrogrammi e spettrogrammi in scala Mel. Per l'addestramento di questi modelli, è stato utilizzato un dataset costituito da un gran numero di registrazioni di urla di donne e bambini, insieme a rumori ambientali per rappresentare situazioni in cui non c'è un pericolo imminente. Lo studio affronta anche il problema del rumore negli audio, applicando varie tecniche di riduzione del rumore per annullarlo o ridurlo, al fine di ottenere risultati più accurati da parte delle reti. Infine, sono state messe a confronto le performance delle reti CNN con le diverse tecniche di riduzione del rumore, con l'obbiettivo di identificare il modello più efficace per individuare quando siamo in presenza di una situazione di pericolo.

## 1.1 Stato dell'arte

Di recente, la sicurezza delle donne e dei bambini è stata oggetto di numerose ricerche, portando allo sviluppo di vari dispositivi e tecnologie volti a prevenire e segnalare situazioni di pericolo. Di seguito, vengono analizzate alcune delle soluzioni più rilevanti sviluppate in questo ambito.
Una delle prime soluzioni proposte è un dispositivo portatile dotato di interruttore a pressione, descritto nella ricerca "Smart Intelligent System for Women and Child Security" [5]. Questo dispositivo permette alla vittima di segnalare una situazione di pericolo premendo un pulsante. Una vota attivato, il dispositivo invia un SMS con la posizione della vittima ai numeri di emergenza preimpostati, seguito da una chiamata. Se la chiamata non riceve risposta entro un certo periodo di tempo, il dispositivo contatta direttamente la polizia. La realizzazione di questo sistema si basa su un microcontrollore Arduino, integrato con diversi moduli, tra cui il GSM800, che permette di effettuare chiamate e inviare messaggi tramite una SIM, e il GPS, utilizzato per determinare la posizione della vittima.
Un'altra soluzione, presentata nell'articolo "Women Safety Device and Application FEMME" [6], propone un dispositivo sincronizzato con un'applicazione Android tramite Bluetooth, ma capace di funzionare anche indipendentemente. Il dispositivo può essere attivato premendo un tasto di emergenza e comprende un modulo GSM per inviare SMS ed effettuare chiamate, un modulo GPS per la localizzazione,

un registratore audio e un rilevatore di telecamere nascoste. Con un singolo click, invia la posizione ed un messaggio di aiuto ai numeri impostati ogni due minuti; con un doppio click, registra l'audio e invia la posizione; con una pressione prolungata, effettua una chiamata alla polizia inviando anche la posizione. Inoltre, il dispositivo è in grado di rilevare telecamere nascoste utilizzando segnali RF (Radio Frequenza). L'app associata, chiamata "FEMME", permette all'utente di scegliere la funzione desiderata tramite interfaccia grafica (messaggio SOS, registrazione audio, rilevatore di telecamere nascoste) oppure di avviare automaticamente l'applicazione premendo determinati tasti, inviando così la posizione e la registrazione ai contati impostati.
Queste soluzioni presentano un limite significativo: richiedono un intervento manuale da parte dell'utente, come la pressione di specifici tasti. Questo può rivelarsi problematico in situazioni di pericolo, in cui l'individuo, trovandosi sotto forte stress o in preda al panico, potrebbe non essere in grado di agire tempestivamente. Per superare questo limite, il paper "Design and development of an IOT based wearable device for the safety and security of women and girl children" [7] ha sviluppato un dispositivo indossabile che monitora continuamente i segnali fisiologici dell'utente, combinandoli con la posizione del corpo per rilevare situazioni di pericolo. Questo dispositivo utilizza l'attività elettrodermica (EDA) e la temperatura corporea come indicatori dello stato emotivo. L'EDA misura le variazioni delle proprietà elettriche della pelle, che cambiano sotto stress a causa dell'aumento della sudorazione, mentre la temperatura corporea tende a diminuire in situazioni di pericolo. Un accelerometro a tre assi valuta la posizione del corpo. I dati raccolti dai sensori vengono inviati in modalità wireless ad un cloud, dove sono analizzati utilizzando MATLAB e algoritmi di machine learning. Questo approccio consente al dispositivo di migliorare la sua precisione nel tempo, grazie all'apprendimento continuo dei dati raccolti.
Un'altra soluzione interessante è rappresentata da applicazioni per dispositivi Android, come descritto nel paper "Lifecraft: An Android Based Application System for Women Safety" [8]. In questo caso, è sufficiente avere uno smartphone con sistema operativo Android, senza la necessità di utilizzare altri dispositivi. Questa applicazione, attivabile tramite un comando vocale o un tasto SOS, invia un messaggio di avviso con la posizione della vittima ai contatti di emergenza ogni cinque

minuti, garantendo un aggiornamento costante della posizione. L'applicazione registra anche l'audio circostante per i primi cinque minuti, fornendo potenziale prove dell'accaduto. L'app funziona sia online che offline, sebbene in modalità offline non sia possibile inviare la posizione GPS, ma solo messaggi di avviso.
Recentemente, le tecnologie di deep learning sono state applicate alla sicurezza di donne e bambini, come illustrato nel paper "Audio signal based danger detection using signal processing and deep learning" [9]. Questa ricerca utilizza reti neurali convoluzionali (CNN) per addestrare modelli capaci di riconoscere situazioni di pericolo in maniera automatica, analizzando urla di donne e bambini. I modelli sono addestrati utilizzando rappresentazioni grafiche degli audio, come gli spettrogrammi di Mel, che permettono di estrarre le caratteristiche rilevanti del segnale audio. Inoltre, la ricerca ha sperimentato l'uso dell'Audio Vision Transformer (Audio ViT), un modello inizialmente sviluppato per la classificazione delle immagini. Nell'Audio ViT, il segnale audio, rappresentato come uno spettrogramma, viene suddiviso in patch, ciascuna delle quali è rappresentata da un token, un vettore numerico che rappresenta una porzione dell'audio. Questi token vengono elaborati dal Transformer, che cattura le relazioni complesse tra le diverse parti del segnale audio, migliorando la capacità del modello di classificare correttamente le situazioni di pericolo. Questa ricerca è stata utilizzata come riferimento nel seguente studio per replicare gli esperimenti, con l'obbiettivo di verificare la validità dei dati presentati e, se possibile, ottimizzare i risultati ottenuti.

# Capitolo 2

# Introduzione alle reti neurali convoluzionali

Ai nostri giorni sentiamo sempre più spesso le parole intelligenza artificiale (AI), che andiamo ad associarla a qualunque tipo di macchina intelligente. In realtà, intelligenza artificiale, machine learning e deep learning sono termini con significati differenti. L'apprendimento automatico è un sottoinsieme dell'intelligenza artificiale. A sua volta, il deep learning è un sottoinsieme del machine learning. In altre parole, tutto il deep learning è machine learning, e tutto il machine learning è intelligenza artificiale (Figura 2.1).
L'intelligenza artificiale è un campo della scienza che si occupa della costruzione di computer e macchine in grado di ragionare, apprendere e agire in modi che solitamente richiederebbero l'intelligenza umana. Il machine learning è un'applicazione dell'AI che consente alle macchine di apprendere senza essere programmate specificamente per un determinato scopo, imparando dalle esperienze precedenti e migliorando attraverso gli errori commessi. Tuttavia, richiede comunque l'intervento umano. L'apprendimento profondo insegna ai computer ad elaborare e analizzare informazioni ispirandosi al funzionamento del cervello umano, correggendosi autonomamente in caso di errore. Il termine deep learning deriva dall'uso di reti neurali profonde, costituite da numerosi strati nascosti di neuroni, simili a quelli del cervello umano. Queste reti richiedono grandi quantità di dati per imparare e risorse hardware significative per eseguire i loro compiti.

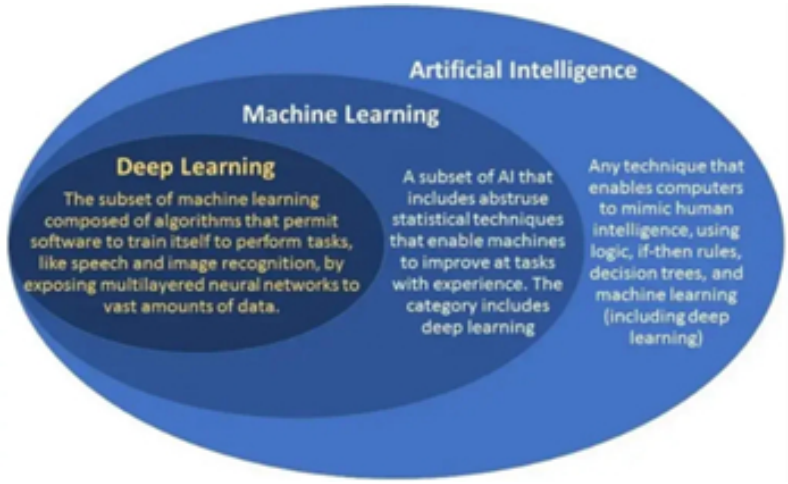



Figura 2.1: Diagramma riassuntivo di Intelligenza Artificiale, Machine Learning e Deep Learning. [10]

## 2.1 Apprendimento nel deep learning

Dopo aver compreso la relazione tra intelligenza artificiale, machine learning e deep learning, è importante analizzare come avviene il processo di apprendimento all'interno di quest'ultimo. Esistono diversi approcci che definiscono il modo in cui un sistema apprende e si adatta ai dati forniti. Questi approcci includono l'apprendimento supervisionato, non supervisionato e per rinforzo. Ciascuna di queste tecniche ha caratteristiche specifiche che la rendono adatta a determinati tipi di problemi e scenari. Di seguito esamineremo le principali modalità di apprendimento:

- Apprendimento Supervisionato, è un tipo di apprendimento che utilizza dati di addestramento 'etichettati' per mappare un input su un output, insegnando all'algoritmo le regole del modello. Per eseguire questo tipo di apprendimento, è necessario un set di dati di addestramento, in cui un 'supervisore' etichetta i dati, associando ogni input al relativo output. Inoltre, ci serve un ulteriore set di dati chiamato di test, solitamente costituito da un sottoinsieme del set di addestramento. L'algoritmo viene addestrato sul set di training, imparando a riconoscere le relazioni tra input e output, che permetteranno poi al modello di fare previsioni. Successivamente, il modello viene testato sul set di test, composto da dati sconosciuti al sistema, confrontando il risultato predetto con quello reale. Se l'errore ottenuto è basso, in base al contesto in cui

stiamo operando, l'addestramento si può considerare concluso, e il modello sarà pronto per operare, ovvero, elaborare nuovi input e fornire output in base a ciò che ha appreso durante l'addestramento.

- Apprendimento Non Supervisionato, è un tipo di apprendimento in cui i dati non sono etichettati e non richiedono un dataset di addestramento. In questo caso, la macchina deve individuare autonomamente le relazioni esistenti tra i dati. Questo tipo di apprendimento è particolarmente utile per identificare modelli nascosti che possono sfuggire all'osservazione umana, sia perché difficili da individuare, sia a causa dell'enorme quantità di dati da analizzare.

- Apprendimento per Rinforzo, non richiede dati etichettati per il condizionamento; è basato sul principio di gratificazione, dove viene fornita una ricompensa per i comportamenti corretti e una punizione per quelli indesiderati. Questo tipo di apprendimento permette al sistema di percepire e interpretare l'ambiente in cui l'agente è immerso, scegliere le azioni da eseguire e imparare dagli errori compiuti.

## 2.2 Reti neurali convoluzionali (CNN)

Le reti neurali cercano di replicare il funzionamento del cervello umano. Il nostro cervello è composto da milioni di neuroni interconnessi che lavorano insieme per apprendere ed elaborare informazioni. Allo stesso modo, le reti neurali sono costituite da molti strati di neuroni artificiali che collaborano tra loro. Questi neuroni artificiali, chiamati nodi, eseguono calcoli matematici per elaborare i dati. Le reti sono organizzate in più livelli di nodi interconnessi, con ciascun livello che perfeziona i risultati del precedente per migliorare progressivamente la previsione. I livelli di input e output sono detti livelli visibili, mentre quelli intermedi sono chiamati livelli nascosti, poiché si trovano all'interno della rete. Il livello di input è dove il modello acquisisce i dati; nei livelli intermedi avviene l'elaborazione, mentre nel livello di output il sistema effettua la previsione (Figura 2.2).

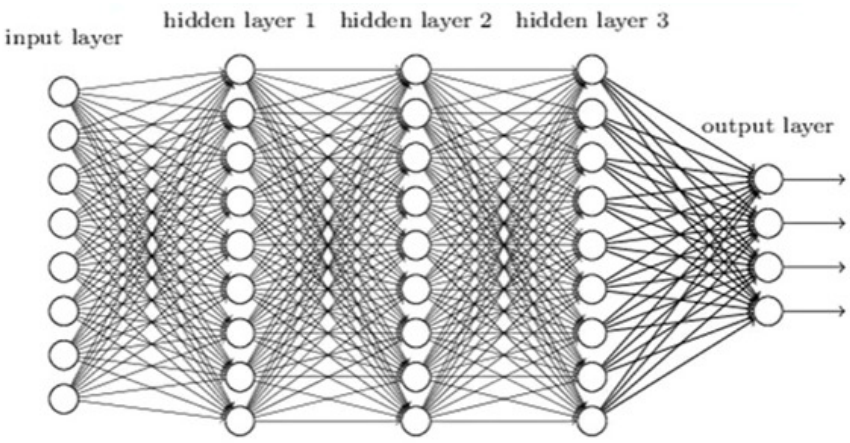



Figura 2.2: Architettura di una rete neurale multistrato. [11]

Ogni nodo all'interno di una rete neurale svolge operazioni matematiche per produrre un output specifico. Comunemente, i dati che un neurone riceve sono rappresentati da vettori o matrici. Ad ogni connessione tra neuroni è associato un peso che ne determina l'importanza, e viene aggiornato durante la fase di addestramento. Questi nodi eseguono una somma del tipo:

$$z = \sum_{i=1}^{n} (\omega_i \cdot x_i) + b \tag{2.1}$$

dove $x_i$ rappresenta gli input, $\omega_i$ i pesi associati e $b$ il bias. Questa somma viene sottoposta ad una funzione di attivazione, che introduce non linearità nel modello, facendo in modo che la rete possa imparare relazione complesse che sussistono tra i dati. L'output ottenuto viene passato poi al successivo neurone dove si ripete il procedimento appena descritto. Le CNN rappresentano i dati in tre dimensioni:

- Larghezza: si riferisce al numero di colonne di pixel presenti nell'immagine, ogni colonna rappresenta una serie di valori di intensità luminosa o colore.
- Altezza: indica il numero di righe di pixel nell'immagine, ogni riga rappresenta come nel caso precedente una serie di valori di intensità luminosa o colore.
- Profondità: rappresenta il numero di canali di colore presenti. Per immagini a colori la profondità solitamente è tre, poiché ogni pixel è rappresentato da tre valori corrispondenti ai canali rosso, verde e blu (RGB). Per quelle in scala di

grigi, la profondità è uno, in quanto ogni pixel è rappresentato da un singolo valore.

Anche se al momento stiamo discutendo delle CNN in relazione all'elaborazione delle immagini, questo concetto può essere esteso anche ai segnali audio. Come vedremo tra poco questo è possibile rappresentando il nostro audio sotto forma di spettrogramma. Descriviamo ora i singoli strati o livelli che compongono la CNN (Figura 2.3).

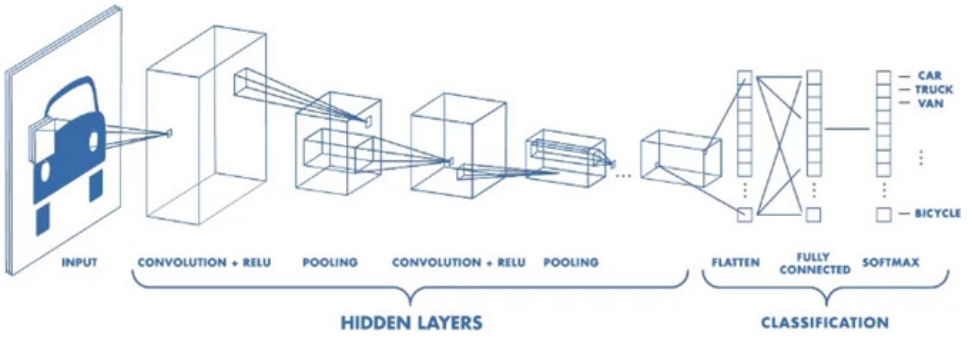



Figura 2.3: Architettura di una rete neurale convoluzionale (CNN). [12]

## 2.3 Strato convoluzionale

Lo strato convoluzionale è la componente principale di una CNN, si occupa di estrarre le caratteristiche dalle immagini di input. Per svolgere questo compito utilizza un insieme di filtri, detti Kernel, piccoli spazialmente (in larghezza e altezza), ma che si estendono per l'intera profondità della dimensione di input. Ad esempio, un tipico filtro potrebbe avere dimensioni $5 \times 5 \times 3$, dove 5 è la larghezza e l'altezza in pixel, e 3 la profondità, corrispondente ai tre canali di colore (RGB) di un'immagine a colori. Questi filtri vengono fatti scorrere su tutto il volume di input, con un passo di avanzamento detto stride, ed eseguono un'operazione di convoluzione (Figura 2.4). Questo processo genera una matrice bidimensionale chiamata feature map o activation map. Tutte le unità di una feature map condividono gli stessi

pesi; quindi, ogni unità è in grado di riconoscere uno specifico pattern in una qualsiasi regione dell'immagine, una volta allenato. A sua volta, ogni feature map, si specializza per riconoscere specifiche caratteristiche e andamenti frequenti in un'immagine. Ogni neurone elabora dati solo per una regione locale del volume di input, chiamata campo ricettivo, che coincide proprio con la dimensione del Kernel.

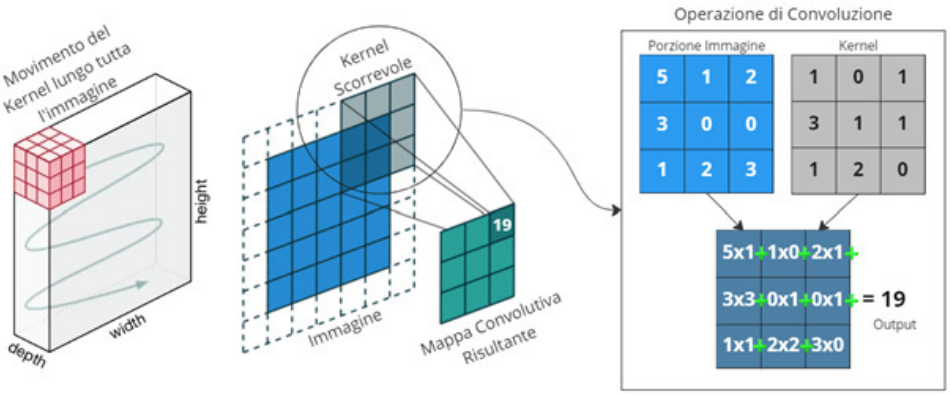



Figura 2.4: Operazione di convoluzione.

Per ogni strato convoluzionale della rete, avremo un insieme di kernel, ognuno dei quali cercherà caratteristiche diverse nell'input, ciascuno dei quali produce una mappa di attivazione. Il numero di filtri utilizzati definisce un parametro detto profondità. Di conseguenza il volume dell'output avrà una profondità pari al numero di filtri utilizzati. Ad esempio, se utilizziamo 64 filtri, l'output sarà una serie di 64 feature map, ognuna delle quali rappresenta una caratteristica rilevata nell'immagine di input. I livelli convoluzionali non vengono applicati solo ai dati di input, ma anche all'output di altri livelli convoluzionali. Questo approccio permette di costruire una gerarchia delle caratteristiche:

- Strati iniziali: i primi strati rilevano caratteristiche di basso livello come bordi e texture semplici.
- Strati intermedi: combinano le caratteristiche di basso livello per riconoscere pattern più complessi.
- Strati profondi: sono in grado di rilevare oggetti complessi come animali, case, ecc.

In alcuni casi si ricorre all'utilizzo del padding, che consiste nel riempire il volume di input con zeri intorno al bordo, questo permette di controllare la dimensione del volume di output.
Se abbiamo un input di dimensioni $W_1 \times H_1 \times D_1$, e $K$ numero di kernel con dimensione spaziale $F$, passo $S$ e padding $P$, produce in volume di uscita di dimensione $W_2 \times H_2 \times D_2$ dove:

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1 \tag{2.2}$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1 \tag{2.3}$$

$$D_2 = K \tag{2.4}$$

## 2.4 Strato ReLu

ReLU, acronimo di "Rectified Linear Unit", è una delle funzioni di attivazione più utilizzate nelle reti neurali, in particolare nelle reti neurali convoluzionali (CNN). Lo strato ReLU viene applicato subito dopo lo strato convoluzionale alle feature map generate dallo strato precedente. Dal punto di vista matematico è definita come:

$$ReLu = max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases} \tag{2.5}$$

dove con $x$ si indica la variabile di input. Questa funzione ha lo scopo di porre a 0 i valori negativi e lasciare inalterati quelli positivi (Figura 2.5), introducendo non linearità nel modello; senza modificare la dimensione dell'output. Ciò consente alla rete di apprendere relazioni più complesse nei dati, migliorando così la sua capacità di generalizzazione e apprendimento.

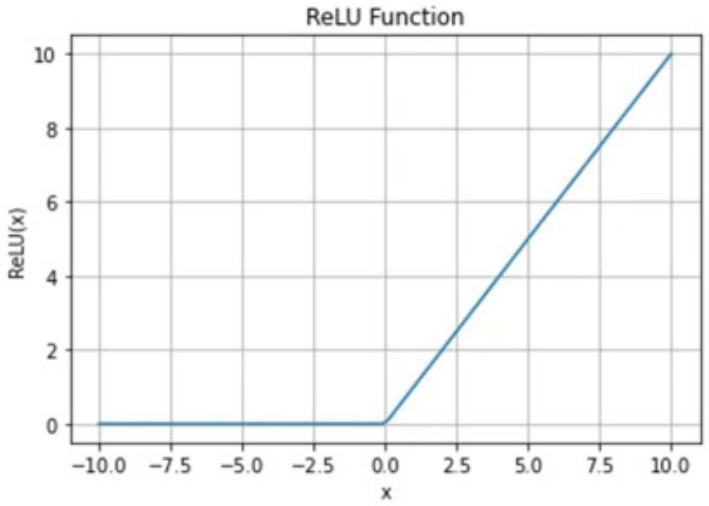



Figura 2.5: Funzione di attivazione ReLu. [13]

## 2.5 Strato di pooling

Solitamente dopo lo strato convoluzionale e quello di ReLu viene messo uno strato di pooling. La sua funzione principale consiste nell'estrarre le caratteristiche più significative dalle feature map, riducendone le dimensioni spaziali. Questo processo utilizza un filtro che scorre su ciascuna mappa di attivazione e produce un output che dipende dal tipo di filtro utilizzato (Figura 2.6). In questo modo si riducono i parametri della rete e, di conseguenza, i calcoli necessari. I principali filtri di pooling sono:

- Max pooling: applica una finestra di dimensioni predefinite, ad esempio $2 \times 2$, su una mappa di attivazione. Questa finestra scorre lungo la mappa con un certo passo (stride), e per ogni posizione, seleziona il valore massimo all'interno della finestra. L'output è una mappa ridotta, dove ogni valore rappresenta il massimo di una regione della mappa originale.

- Average pooling: a differenza del Max pooling, l'average pooling calcola la media dei valori presenti nella finestra. L'output è una mappa ridotta, dove ogni valore rappresenta il valore medio di una regione della mappa originale.

- L2 pooling: in questo caso si calcola la noma L2 (o norma Euclidea) sui valori presenti nella finestra. L'output è una mappa ridotta, dove ogni valore rappresenta la norma Euclidea di una regione della mappa originale.

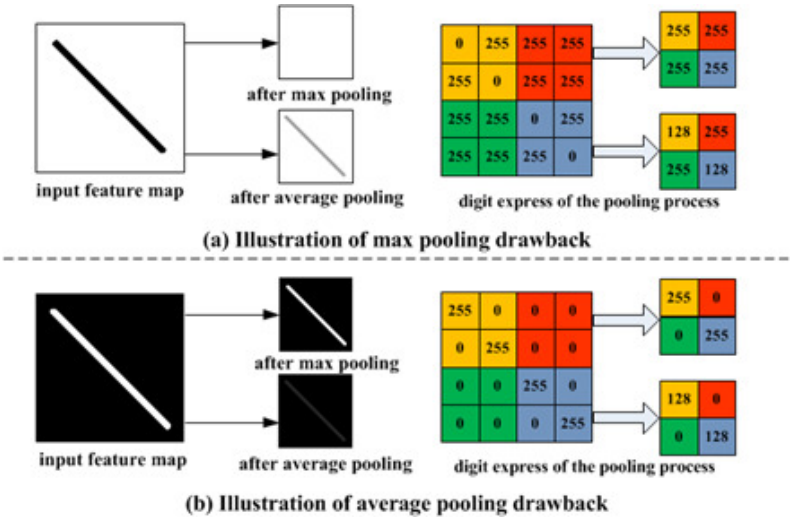



Figura 2.6: Rappresentazione Max pooling e Average pooling. [14]

Per un input di dimensione $W_1 \times H_1 \times D_1$, con dimensione spaziale $F$ e passo $S$, otteniamo un volume di uscita $W_2 \times H_2 \times D_2$ con:

$$W_2 = \frac{(W_1 - F)}{S} + 1 \tag{2.6}$$

$$H_2 = \frac{(H_1 - F)}{S} + 1 \tag{2.7}$$

$$D_2 = D_1 \tag{2.8}$$

Per i livelli di pooling, non è comune usare lo zero-padding sull'input.

## 2.6 Strato completamente connesso

Uno strato completamente connesso, noto anche come strato denso, è un tipo di strato in cui ogni neurone è collegato a tutti i neuroni dello stato precedente. Si trovano generalmente alla fine dell'architettura di una rete neurale, dopo i livelli convoluzionali e di pooling, e sono responsabili della produzione delle previsioni sull'output finale.

Prima di essere elaborato da uno strato completamente connesso, l'output multidimensionale proveniente dagli strati convoluzionali e di pooling viene appiattito (flattened) in un vettore unidimensionale. Questo processo di appiattimento converte una struttura bidimensionale o tridimensionale in una singola lista di valori, rendendo i dati compatibili con lo strato completamente connesso.
Durante l'allenamento, i pesi e i bias in questo strato vengono appresi e adattati al problema specifico. Il numero di neuroni in questo strato corrisponde al numero di classi di output in un problema di classificazione, permettendo alla rete di assegnare una probabilità di appartenenza a ciascuna classe.

## 2.7 Strato di loss

La funzione Softmax è una funzione di attivazione utilizzata principalmente negli stati finali delle reti neurali per problemi di classificazione multiclasse. Essa trasforma un vettore z di K numeri reali in un vettore di probabilità, dove la somma di tutte le probabilità è uguale a uno. Queste probabilità saranno poi usate per determinare la classe predetta dal modello. Matematicamente, la funzione softmax per un vettore di input è definita come:

$$\sigma : \mathbb{R}^K \to \left\{ z \in \mathbb{R}^K \mid z_i > 0, \sum_{i=1}^{K} z_i = 1 \right\} \tag{2.9}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \texttt{per j=1,...,K} \tag{2.10}$$

# Capitolo 3

# Modelli utilizzati

In questo studio, sono state utilizzate diverse reti neurali convoluzionali (CNN) per la classificazione audio. La scelta di impiegare questa tipologia di reti deriva dalla loro eccellente capacità di estrarre caratteristiche significative dai dati, come le immagini. Tuttavia, anziché utilizzare gli audio in forma grezza, questi sono stati trasformati in spettrogrammi Mel, una rappresentazione visiva delle frequenze sonore nel tempo (di cui si parlerà successivamente). Questa conversione consente al modello di trattare gli spettrogrammi come immagini bidimensionali, permettendo alle reti di identificare ed estrarre le caratteristiche più rilevanti per la classificazione. Le reti utilizzate sono:

- InceptionV3
  - Molto efficace per la classificazione di immagini complesse.
- MobileNetV2
  - Modello leggere ed efficiente.
  - Particolarmente adatto per dispositivi mobili e applicazioni con risorse limitate.
- ResNet50
  - Modello profondo con 50 strati convoluzionali.
  - Ottimo per la classificazione di immagini con un elevato numero di classi.

- ResNet101
  - Versione più profonda di ResNet50 con 101 strati convoluzionali.
  - Ancora più preciso di ResNet50, ma richiede più tempo e risorse computazionali per l'addetramento.
- Xception
  - Molto efficace per la segmentazione e la classificazione di immagini con oggetti di dimensioni variabili.

L'impiego di queste reti con diverse architetture e caratteristiche consente di ottenere una panoramica più completa delle loro prestazioni nella classificazione audio. In particolare, la varietà dei modelli permette di valutare la capacità di ciascuna rete nell'estrarre e riconoscere le caratteristiche saliente dai dati audio trasformati in spettrogrammi Mel.

## 3.1 InceptionV3

InceptionV3 [4] è un modello di deep learning basato su reti neurali convoluzionali, utilizzato per la classificazione delle immagini. La prima versione di questa rete è l'InceptionV1. Questo modello per prevenire un adattamento eccessivo dei dati (overfitting) causato dall'utilizzo di più strati convoluzionali, impiega filtri di diverse dimensioni sullo stesso livello. Pertanto, anziché svilupparsi in profondità, il modello presentava strati paralleli, rendendolo più ampio che profondo. Il modulo base dell'InceptionV1 (chiamato modulo Inception, Figura 3.1) è costituito da quattro layer paralleli:

1. $1 \times 1$ Convolution
2. $3 \times 3$ Convolution
3. $5 \times 5$ Convolution
4. $3 \times 3$ Max pooling

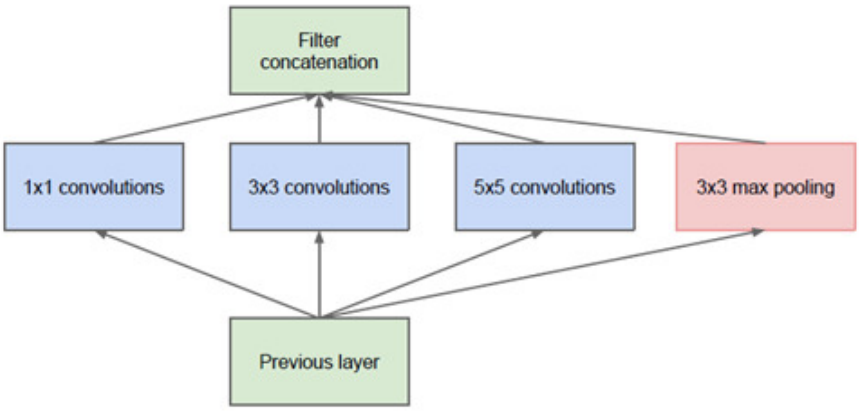



Figura 3.1: Modulo Inception. [15]

Questo modulo esegue la convoluzione su un input utilizzando filtri di tre diverse dimensioni e include anche un'operazione di max pooling. Gli output risultanti vengono concatenati e passati al modulo successivo. Uno degli svantaggi di questa struttura è l'uso della convoluzione $5 \times 5$, che è computazionalmente costosa, richiedendo un'elevata potenza di calcolo e tempi di elaborazione più lunghi. Per ovviare a questo problema, è stato introdotto uno strato di convoluzione $1 \times 1$ prima di ogni strato convoluzionale, riducendo così le dimensioni della rete e velocizzando i calcoli (Figura 3.2).

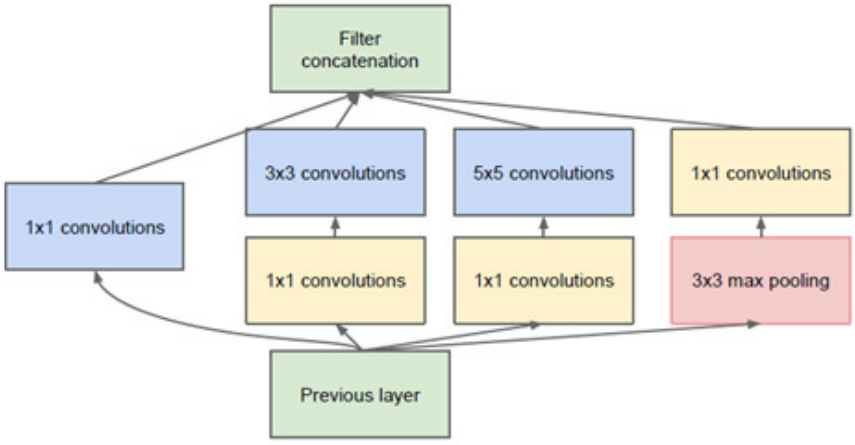



Figura 3.2: Modulo Inception con riduzione delle dimensioni. [15]

Utilizzando il modulo Inception a dimensioni ridotte, è stata sviluppata una rete neurale nota come InceptionV1, caratterizzata da una profondità di 22 livelli (o 27 se si considerano anche gli strati di pooling).
Il modello V3 (Figura 3.3), con i suoi 42 livelli, rappresenta un'evoluzione rispetto al predecessore V1, sviluppato da Google nel 2014 come parte dell'architettura GoogLeNet. Una delle principali innovazioni di InceptionV3 è la fattorizzazione delle convoluzioni. Invece di utilizzare direttamente strati convoluzionali $5 \times 5$, questi sono stati suddivisi in due strati $3 \times 3$. Questo approccio ha ridotto significativamente il numero di parametri e i costi computazionali, migliorando l'efficienza complessiva della rete. Inoltre, è stata introdotta la fattorizzazione spaziale tramite convoluzioni asimmetriche, che prevede la sostituzione delle convoluzioni $3 \times 3$ con una combinazione di convoluzioni $1 \times 3$ e $3 \times 1$. Questo consente di ottenere lo stesso campo recettivo delle convoluzioni $3 \times 3$ ma con un minor numero di parametri. Un'altra importante innovazione di InceptionV3 è l'introduzione dei classificatori ausiliari, progettati per migliorare la convergenza nelle reti neurali profonde e affrontare il problema del gradiente evanescente.
Il gradiente evanescente è un problema che si verifica durante il processo di addestramento e ostacola l'apprendimento efficace della rete. Nelle reti molto profonde, i gradienti che guidano l'aggiornamento dei pesi possono diminuire man mano che vengono propagati all'indietro, specialmente negli strati iniziali. Se i gradienti diventano troppo piccoli, i pesi degli strati iniziali non vengono aggiornati in modo efficace, impedendo alla rete di apprendere correttamente. I classificatori ausiliari sono strati aggiuntivi posizionati strategicamente all'interno della rete per generare una funzione di perdita ausiliaria. Questo approccio aiuta a mantenere i gradienti significativi anche negli strati iniziali, facilitando l'aggiornamento dei pesi lungo la rete e migliorando sia la capacità di apprendimento che la velocità di convergenza del modello.

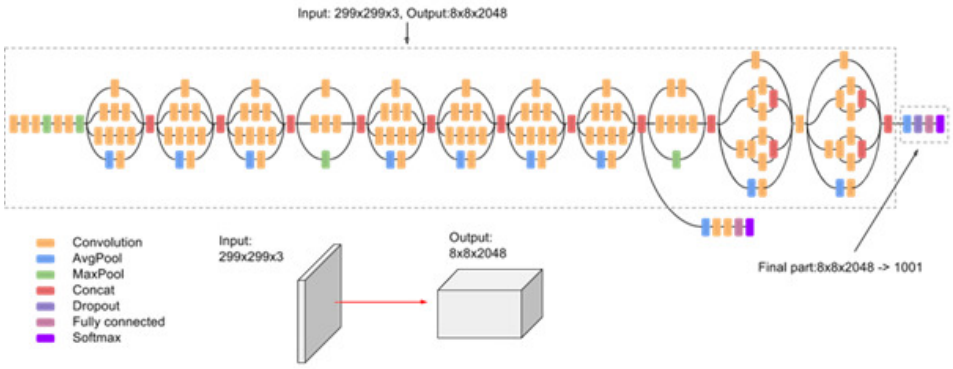



Figura 3.3: Architettura rete InceptionV3. [15]

## 3.2 MobileNetV2

MobileNet [3] è un'architettura di rete neurale sviluppata da Google nel 2017, progettata specificamente per dispositivi mobili e ambienti con risorse computazionali limitate. La caratteristica distintiva di questo modello è la sua elevata velocità di elaborazione, combinata con prestazioni eccellenti in termini di accuratezza. La prima versione è la MobileNetV1, le principali caratteristiche sono:

- Convoluzione separabile in profondità (Depthwise separable convolution).
- ReLu 6.

La convoluzione separabile in profondità (Figura 3.6), riduce significativamente il costo computazionale rispetto alle convoluzioni standard. Essa è composta da due operazioni: la convoluzione depthwise e la convoluzione pointwise. Nella convoluzione depthwise (Figura 3.4), invece di applicare un filtro su tutti i canali di input, come avviene in quella standard, questa convoluzione applica un filtro separato a ciascun canale. Ad esempio, se abbiamo un'immagine a tre canali di colore (RGB) e un filtro, la convoluzione divide sia il filtro che l'immagine in tre canali distinti, esegue le convoluzioni su ciascun canale in modo indipendente, e successivamente ricompone i canali.

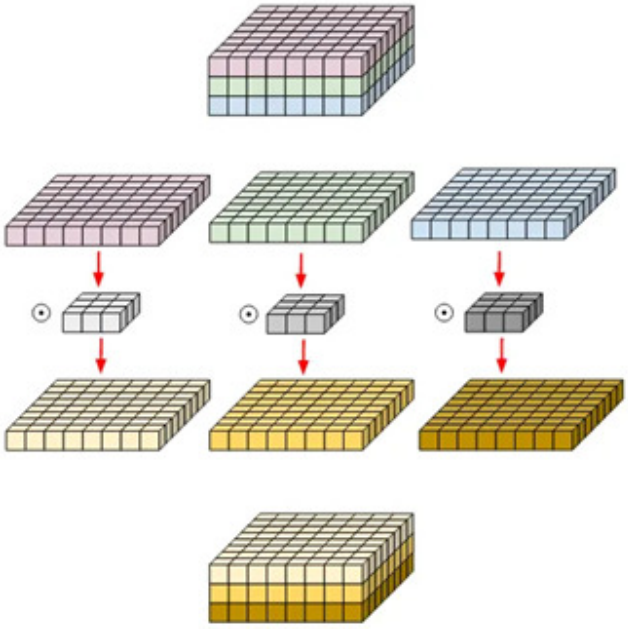


Figura 3.4: Depthwise convolution. [16]

Questa operazione permette di ottenere un'uscita a tre canali usando un solo filtro per ciascun canale, mentre una convoluzione normale richiederebbe tre filtri per i tre canali.
La convoluzione pointwise (Figura 3.5), consiste in una convoluzione $1 \times 1$, utilizzata per combinare le mappe di caratteristiche ottenute da quella precedente. Questa operazione combina i risultati dei diversi canali in una singola mappa di caratteristiche.

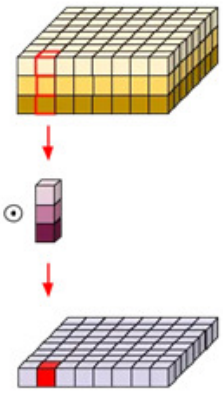


Figura 3.5: Pointwise convolution. [16]

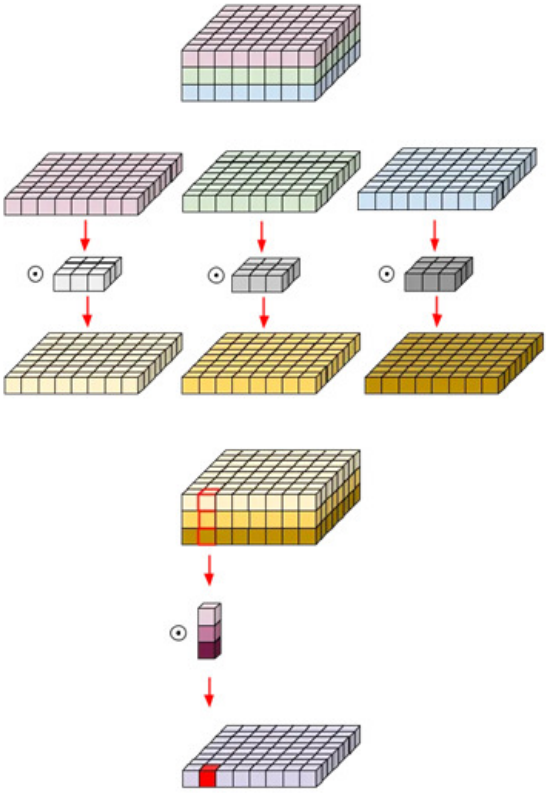


Figura 3.6: Depthwise separable convolution (formata dalla Depthwise convolution e dalla Pointwise convolution). [16]

La funzione ReLu 6 introduce la non linearità e limita l'output massimo a 6. Questa limitazione rende i calcoli più efficienti e veloci. Dal punto di vista matematico è

definita come:

$$ReLu6(x) = min(max(0, x), 6) \tag{3.1}$$

La versione V2 migliora la versione precedente introducendo due nuove caratteristiche il collo di bottiglia lineare (Linear bottleneck) e il blocco residuo invertito (Inverted residual block). Con il blocco residuo invertito, l'input passa in una convoluzione pointwise che espande il numero di canali, per catturare più caratteristiche. I canali espansi vengono elaborati tramite una convoluzione depthwise, che è computazionale efficiente. L'output viene poi compresso (collo di bottiglia lineare) di nuovo usando una convoluzione pointwise che riduce il numero di canali. In questo passaggio non si applica alcuna funzione ReLu, per preservare meglio le informazioni originali. Infine, utilizziamo la connessione residuale, con cui si collega l'output con l'input originale (se le dimensioni sono compatibili). Questa connessione migliora la capacità di apprendimento del modello e riduce la possibilità di dispersione del gradiente. Questo approccio consente a MobileNetV2 di ottenere una maggiore efficienza computazionale e migliori prestazioni su dispositivi mobili.

## 3.3 ResNet50

La ResNet (o Residual Network) è un'architettura di rete neurale convoluzionale profonda, introdotta da Microsoft Research nel 2015. Questo modello, noto per la sua eccellente performance nella classificazione delle immagini, è progettato per essere addestrato su grandi dataset, offrendo risultati altamente affidabili. In questo studio, è stata utilizzata la ResNet50 [1] (Figura 3.7), che prende il nome dal fatto che la rete è composta da 50 livelli (48 strati convoluzionali, 1 strato di MaxPooling e 1 strato di AveragePooling). L'esigenza di un modello di questo tipo deriva dalla necessità di affrontare il problema del gradiente evanescente, che si manifesta nelle reti neurali profonde. Questo problema, già discusso in relazione alle reti Inception, comporta che i gradienti diventino molto piccoli man mano che vengono propagati all'indietro, rendendo difficile l'aggiornamento efficace dei pesi negli strati iniziali. La ResNet risolve questo problema introducendo le Skip Connection o connessioni residue. È composta da quattro blocchi principali:

- Strati convoluzionali.

- Blocco identità.
- Blocco convoluzionale.
- Strati completamente connessi.

Gli strati convoluzionali sono responsabili dell'estrazione delle caratteristiche dall'immagine di input. Applicano una serie di filtri all'immagine, producendo delle feature map. Gli strati convoluzionali sono seguiti da una normalizzazione batch, dall'attivazione ReLu e da uno strato di max pooling.
Il blocco identità e il blocco convoluzionale sono gli elementi chiave di ResNet50. Il blocco identità viene utilizzato per garantire che l'input e output abbiano le stesse dimensioni. È costituito da tre stati convoluzionali, ciascuno seguito da funzioni di normalizzazione batch e di attivazione ReLu. L'input originale viene sommato all'output del terzo strato convoluzionale, permettendo una facile propagazione dei gradienti e migliorando la capacità di apprendimento della rete. Il blocco convoluzionale è simile al blocco identità, con l'aggiunta di uno strato convoluzionale $1 \times 1$, utilizzato per modificare il numero di canali dell'input, rendendo compatibili le dimensioni di input e output.
La parte finale sono i livelli completamente connessi, il cui obbiettivo è eseguire la classificazione. L'output delle convoluzioni viene appiattito e passato attraverso una serie di strati densi, terminando con una funzione softmax per produrre la probabilità delle classi predette.

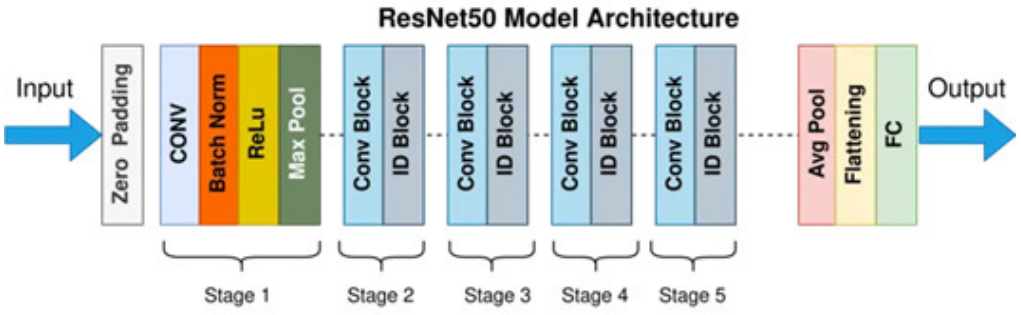



Figura 3.7: Architettura rete ResNet50. [17]

Abbiamo accennato che il problema del gradiente evanescente è stato risolto con le skip connection (Figura 3.8). Queste connessioni, utilizzate sia nel blocco identità sia in quello convoluzionale, permettono alle informazioni di fluire direttamente

dall'input all'output, saltando uno o più strati intermedi. Questo meccanismo consente alla rete di apprendere funzioni residue che mappano l'input sull'output desiderato, migliorando così l'efficacia dell'addestramento e facilitando una più efficace propagazione del gradiente attraverso la rete profonda.

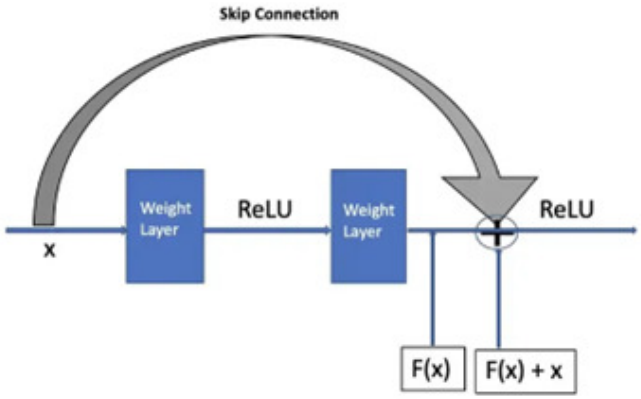



Figura 3.8: Skip connection. [17]

## 3.4 ResNet101

La ResNet101 [1] (Figura 3.9), ha le stesse caratteristiche elencate della ResNet50, in quanto entrambe fanno parte della famiglia delle reti ResNet (Residual Network) introdotte da Microsoft. La ResNet101 può essere considerata un'evoluzione della versione a 50 strati. La differenza principale è nel numero di strati: la ResNet101 è composta da 101 strati, rispetto ai 50 della ResNet50. Questa maggiore profondità consente di catturare dettagli più complessi dai dati di input, migliorando la precisione della classificazione.
Gli strati convoluzionali iniziali, il blocco identità, il blocco convoluzionale e gli strati completamente connessi seguono la stessa logica della ResNet50, con la differenza della maggiore profondità della ResNet101. Inoltre, anche in questo caso il blocco identità e quello convoluzionale usano le skip connection per bypassare uno o più strati, risolvendo il problema del gradiente evanescente.

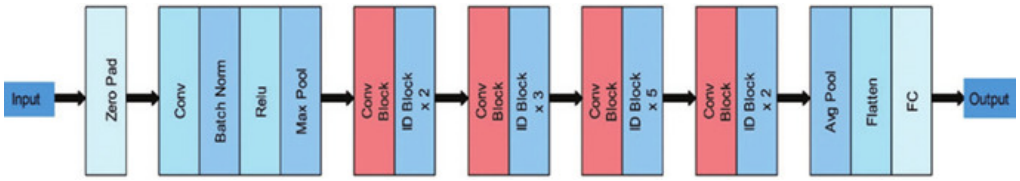



Figura 3.9: Architettura rete ResNet101. [18]

## 3.5 Xception

La Xception [2] (Extreme Inception), (Figura 3.10), è un'architettura di rete neurale convoluzionale introdotta da Francois Cholette, l'autore della libreria Keras. È un'estensione del modello Inception, che sfrutta il concetto di convoluzioni separabili in profondità (depthwise separable convolutions). Il modello Xception è progettato per essere più efficiente e potente rispetto ai suoi predecessori, migliorando le prestazioni su diversi compiti di classificazione delle immagini. La caratteristica principale è l'utilizzo delle convoluzioni separabili in profondità. Le convoluzioni tradizionali operano simultaneamente sia sulla dimensione spaziale che su quella di profondità. Quelle separabili in profondità, come accennato in precedenza, sono formate da due operazioni: convoluzione depthwise con cui si applica un filtro separato a ciascun canale di input, e convoluzione pointwise utilizza una convoluzione $1 \times 1$ per combinare i canali di input risultanti dalla convoluzione depthwise. In questo caso la depthwise separable convolution, viene modificata, nel senso che si esegue prima la pointwise convolution e in seguito la depthwise convolution. Usando queste convoluzioni si riduce significativamente il numero di parametri ed i costi computazionali.
La rete Xception è composta da 36 blocchi di convoluzione ripetuti che formano la base della rete. L'architettura può essere suddivisa in tre parti principali:

- Flusso di ingresso (Entry), include pochi strati convoluzionali tradizionali per estrarre caratteristiche a basso livello e ridurre le dimensioni dell'immagine, dopo l'estrazione delle feature a basso livello, si applicano gli strati convoluzionali separabili in profondità. Vengono applicate anche funzioni ReLu dopo i primi strati convoluzionali tradizionali e come ultimo livello abbiamo la funzione MaxPooling per ridurre le dimensioni delle feature map.

- Flusso medio (Middle), rappresenta la parte più significativa della rete. Questo blocco viene ripetuto otto volte. Ogni ripetizione è composta da strati convoluzionali separabili in profondità e funzioni di attivazione ReLu. Ripetendolo otto volte, il flusso centrale estrae progressivamente dall'immagine le caratteristiche di livello superiore.

- Flusso di uscita (Exit), include strati convoluzionali separabili in profondità, uno strato di AveragePooling e uno strato completamente connesso per la classificazione.

Questo modello utilizza anche le skip connection in tutta la sua architettura, per migliorare il flusso di informazioni attraverso la rete e prevenire il problema del gradiente evanescente. La combinazione di convoluzioni separabili in profondità e delle connessioni residue rende la Xception una delle reti neurali convoluzionali più efficienti, sia in termini di computazione che di prestazioni.

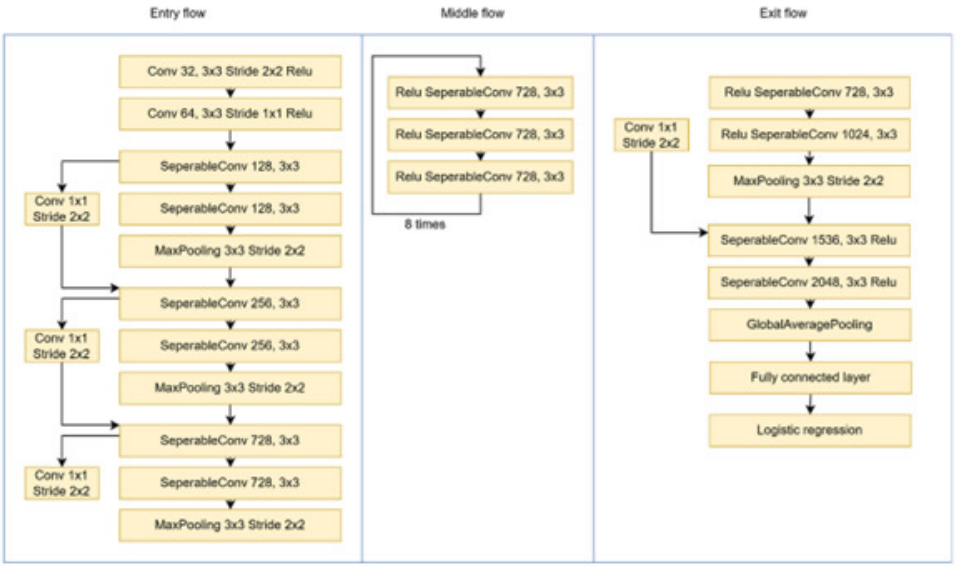



Figura 3.10: Architettura rete Xception. [19]

# Capitolo 4

# Teoria degli strumenti

In questa sezione, discuterò brevemente sulla natura del suono, analizzando le sue caratteristiche e le sue rappresentazioni nel contesto dell'elaborazione audio. Successivamente, esaminerò come i suoni vengono trasformati in spettrogrammi, e in spettrogrammi Mel, una rappresentazione utile per l'addestramento dei modelli di deep learning. Verranno poi presentate le principali tecniche di riduzione del rumore, fondamentali per migliorare la qualità dei dati audio e, di conseguenza, le prestazioni dei modelli. Infine, approfondirò le metriche di valutazione impiegate per misurare l'efficacia e l'accuratezza delle reti neurali utilizzate nella classificazione dei suoni.

## 4.1 Cos'è un suono

Il suono è una vibrazione che si propaga sotto forma di onde meccaniche attraverso un mezzo, come l'aria, l'acqua o i solidi. Queste vibrazioni sono percepite dall'orecchio umano come suoni, e vengono analizzate ed interpretate dal cervello. Le caratteristiche di un suono sono:

- Frequenza, si riferisce al numero di oscillazioni o cicli che un'onda sonora compie in un secondo, misurata in (Hz). Determina l'altezza del suono: alte frequenze corrispondono a suoni acuti, basse frequenze a suoni gravi.
- Ampiezza di un'onda sonora rappresenta la sua intensità o volume. Maggiore è l'ampiezza, più forte sarà il suono percepito. Viene misurata in decibel (dB).

- Timbro è la qualità distintiva di un suono che lo rende riconoscibile e unico, anche se ha la stessa frequenza ed ampiezza di un altro. È influenzato dalla forma d'onda e dalle armoniche del suono, permettendo di distinguere, ad esempio la voce di una persona da un'altra o il suono di un pianoforte da quello di un violino.

- Fase, descrive la posizione dell'onda sonora rispetto a un punto di riferimento. Differenze di fase possono influenzare come i suoni si combinano tra loro.

Per analizzare e processare il suono, questo viene spesso convertito in rappresentazioni grafiche come:

- Forma d'onda (Figura 4.1), una rappresentazione del segnale nel dominio del tempo, che mostra come l'ampiezza del segnale varia nel tempo.

- Spettrogramma (Figura 4.6), una rappresentazione tridimensionale che mostra come l'energia del segnale varia nel tempo e nelle frequenze.

- Spettrogramma Mel (Figura 4.10), una versione dello spettrogramma in cui le frequenze sono mappate su una scala Mel, che rispecchia più fedelmente la percezione umana delle frequenze.

Queste rappresentazioni visive consentono di osservare le caratteristiche temporali e frequenziali del suono, facilitando l'interpretazione e il trattamento del segnale audio, soprattutto in ambiti di elaborazione del segnale e machine learning.

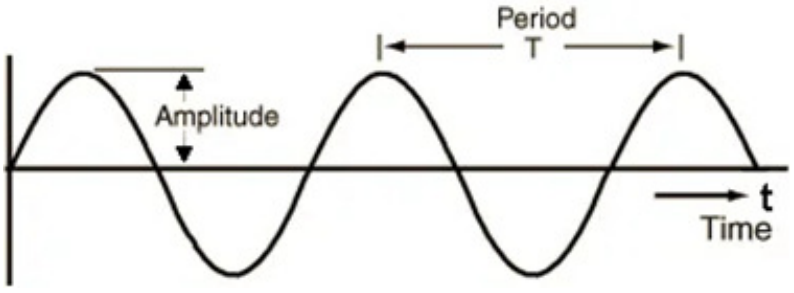



Figura 4.1: Forma d'onda di un segnale. [20]

La maggior parte dei segnali che incontriamo non presenta andamenti semplici e regolari. Questo accade perché segnali di diverse frequenze possono essere sommati per creare segnali compositi con schemi più complessi. Tutti i suoni che sentiamo, inclusa la voce umana, sono costituiti da tali forme d'onda complesse. Di seguito è riportato un esempio di forma d'onda di un segnale tratto dal dataset utilizzato in questo studio. Come possiamo osservare, il segnale presenta un andamento più complesso rispetto alla semplice forma d'onda mostrata nella Figura 4.1.

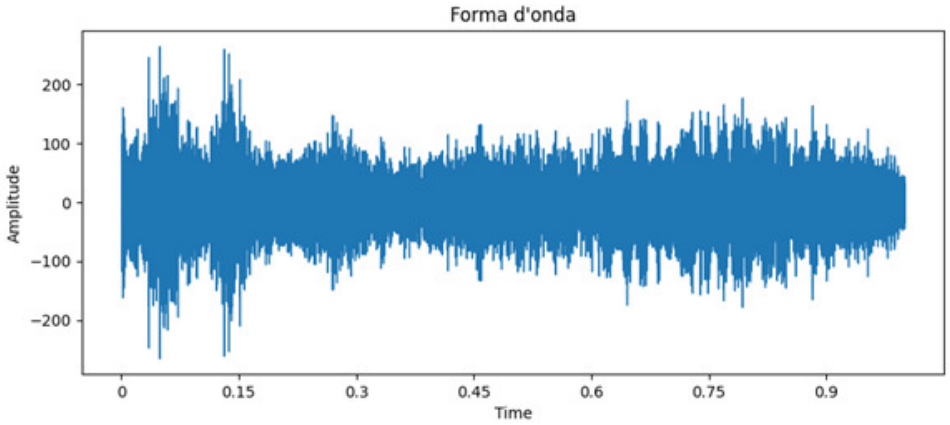



Figura 4.2: Esempio di forma d'onda di un audio del dataset.

## 4.2 Digitalizzazione del suono

Il segnale continuo che varia nel tempo deve essere convertito in una serie di valori discreti per essere elaborato da un computer. Questo processo, noto come digitalizzazione del suono (Figura 4.3), comporta due passaggi principali: il campionamento e la quantizzazione.

Il campionamento è il processo di misurazione del segnale a intervalli regolari di tempo. Questo trasforma il segnale continuo in una sequenza di valori discreti. La frequenza di campionamento (sample rate) è il numero di campioni prelevati per secondo, misurata in Hertz (Hz). Una frequenza comune per l'audio di alta qualità è 44.1 kHz, che significa 44.100 campioni al secondo. Nel seguente studio gli audio sono stati campionati con un sample rate di 40 kHz, ovvero, 40.000 campioni al secondo.

La quantizzazione è il processo di mappatura di ciascun campione del segnale continuo a un valore discreto all'interno di un intervallo finito. Una caratteristica importante è la profondità di bit (bit depth), ovverosia il numero di livelli discreti che possono essere rappresentati. Una maggiore profondità consente una rappresentazione più precisa della variazione di ampiezza del segnale. Le profondità di bit più comuni sono 16 e 24 bit. Ciascun bit rappresenta un termine binario, che indica il numero di livelli discreti a cui il valore dell'ampiezza può essere quantizzato durante la conversione da continuo a discreto. Poiché questo processo implica l'arrotondamento del valore continuo a un valore discreto, si introduce inevitabilmente un certo rumore.

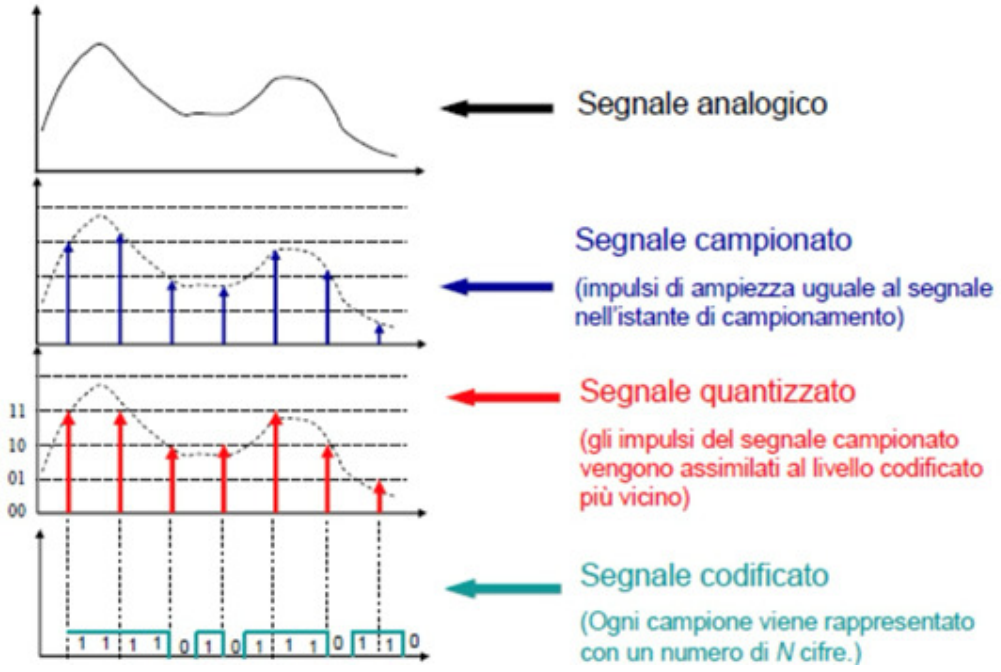



Figura 4.3: Digitalizzazione del suono.

Ora iniziamo a vedere come trasformare il nostro segnale in un'immagine, così da poterla utilizzare con le reti neurali convoluzionali per la classificazione. Per fare questo, introduciamo i concetti di spettrogramma e spettrogramma Mel.

## 4.3 Spettrogramma

Prima di esplorare il concetto di spettrogramma, è fondamentale comprendere cosa sia uno spettro. Abbiamo già accennato che ogni suono che si verifica nel mondo reale può essere considerato come una combinazione di molteplici frequenze. Questo significa che qualsiasi segnale può essere rappresentato come la somma di diverse frequenze distinte, ognuno con la propria ampiezza.
Lo spettro di un segnale è una rappresentazione delle frequenze che compongono quel segnale, insieme alle ampiezze associate a ciascuna frequenza (Figura 4.4). In altre parole, lo spettro traccia tutte le frequenze presenti nel segnale e indica quanto è intensa (o ampia) ogni frequenza.

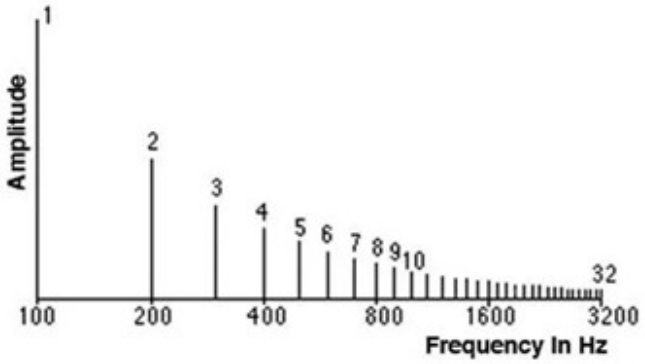



Figura 4.4: Spettro che mostra le frequenze che compongono un segnale. [20]

La frequenza più bassa prende il nome di frequenza fondamentale. Le altre che sono multipli della frequenza fondamentale sono chiamate armoniche.
Abbiamo già visto la forma d'onda di un suono, dove l'asse $x$ rappresenta il tempo e l'asse $y$ rappresenta l'ampiezza. Questa è una rappresentazione del segnale nel dominio del tempo. Lo spettro, invece, è un modo alternativo per rappresentare lo stesso segnale. In questo caso, l'asse $x$ rappresenta le frequenze e l'asse $y$ l'ampiezza di ciascuna frequenza. Questa è una rappresentazione del segnale nel dominio della frequenza. Per passare dalla rappresentazione nel dominio del tempo alla rappresentazione nel dominio delle frequenze si utilizza la trasformata di Fourier discreta (DFT), (Figura 4.5).

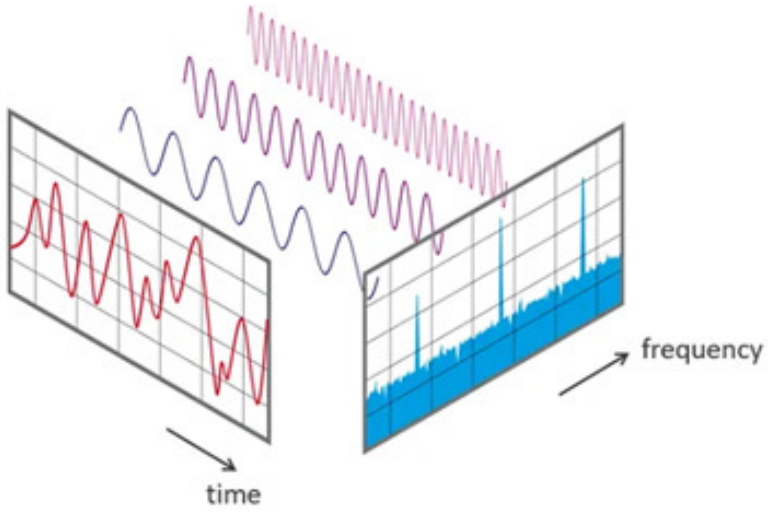



Figura 4.5: Rappresentazione dominio del tempo e dominio della frequenza. [20]

Lo spettrogramma è una rappresentazione visiva delle frequenze di un segnale nel tempo, possiamo dire che è una "fotografia" del segnale. Mostra come lo spettro delle frequenze di un segnale varia nel tempo. È un grafico, in cui sono riportate le frequenze che compongono l'onda sonora al passare del tempo, sull'asse $x$ abbiamo il tempo, sull'asse $y$ le frequenze. Utilizza colori diversi per indicare l'ampiezza di ciascuna frequenza, più luminoso è il colore maggiore è l'energia del segnale (Figura 4.6). Ciascuna "fetta" verticale dello spettrogramma è essenzialmente lo spettro del segnale in un dato istante di tempo. Questa "fetta" mostra come l'intensità del segnale è distribuita tra le diverse frequenze in quell'istante.

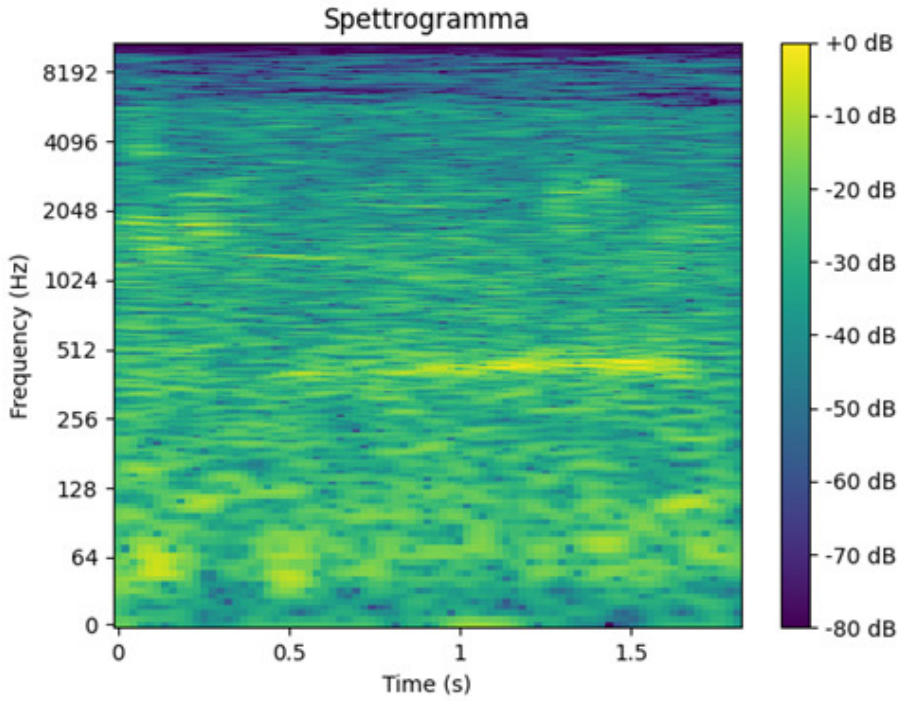



Figura 4.6: Spettrogramma.

Uno spettrogramma viene ottenuto utilizzando la Short-Time Fourier Transform (STFT). Con STFT si indica una tecnica utilizzata per analizzare segnali non stazionari, cioè segnali le cui caratteristiche cambiano nel tempo. La STFT a tempo discreto è definita come:

$$STFT\left\{x[n]\right\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n-m]e^{-i\omega n} \tag{4.1}$$

dove $x[n]$ rappresenta il segnale che si desidera analizzare nel dominio tempo-frequenza e con $\omega[n]$ indichiamo la funzione finestra applicata al segnale.
La procedura per calcolare la STFT consiste nello scomporre il segnale in segmenti di breve durata, dette finestre temporali, della durata di pochi millisecondi, generalmente in un intervallo che va dai $5ms$ ai $10ms$. Ogni finestra viene sovrapposta alla successiva per catturare la continuità del segnale, una sovrapposizione comune è del 50%. Successivamente, si applica la Fast Fourier Transform (FFT) a ciascuna finestra per ottenere lo spettro di frequenza di quel segmento (Figura 4.7). La FFT è un algoritmo per calcolare la trasformata di Fourier discreta (DFT), utilizzata per

convertire un segnale dal dominio del tempo al dominio delle frequenze. È definita come:

$$X_q = F_d(x_n) = \sum_{k=0}^{N-1} x_k e^{-i\frac{2\pi}{N}kq} \qquad q = 0, \ldots, N-1. \tag{4.2}$$

in cui $x_k$ rappresenta il $k-esimo$ campione del segnale $x_n$, il termine esponenziale complesso $e^{-i\frac{2\pi}{N}kq}$ rappresenta la base della trasformata di Fourier discreta e serve a decomporre il segnale nei suoi componenti di frequenza, infine con la $q$ che varia tra 0 e $N-1$ si indica l'indice di frequenza discreta. I risultati delle FFT di tutti i segmenti vengono poi combinati per ottenere la rappresentazione tempo-frequenza (spettrogramma). Generalmente l'asse delle ordinate viene convertito in una scala logaritmica e l'ampiezza viene convertita in decibel, dB.

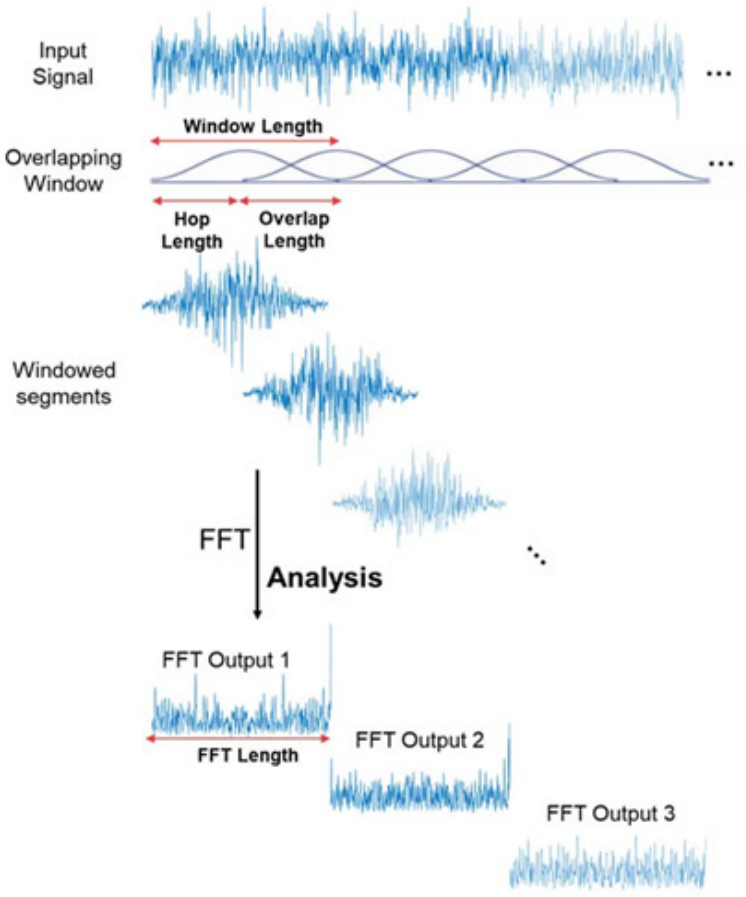



Figura 4.7: Calcolo dello spettrogramma utilizzando la STFT. [21]

## 4.4 Spettrogramma di Mel

La scala Mel, il cui nome deriva da "melody", è una scala percettiva dell'altezza di un suono che nasce dallo studio della percezione del suono da parte dell'orecchio umano. Fu proposta da Stevens, Volkman e Newman nel 1937. La scala è il risultato di esperimenti psico-acustici in cui ai partecipanti è stato chiesto di ascoltare due toni (sinusoidali) con frequenze diverse e chiedere loro di regolare uno dei toni fino a percepirlo come il doppio o la metà del tono di riferimento. Gli esperimenti hanno dimostrato che l'orecchio umano percepisce le differenze di frequenza in modo non lineare. Ad esempio, l'intervallo percepito tra $100Hz$ e $200Hz$ è molto maggiore rispetto a quello tra $1000Hz$ e $1100Hz$, anche se la differenza in termini di Hz è la stessa ($100Hz$). Questo perché l'orecchio umano è più sensibile alle differenze nelle basse frequenze rispetto quelle ad alte frequenze. Da questi risultati i ricercatori hanno derivato una formula che mappa le frequenze in Hz con i valori della scala Mel. Il punto di riferimento tra questa scala e la normale misurazione della frequenza è definito assegnando un'altezza percettiva di $1000Mels$ a un tono di $1000Hz$. Esistono diverse formulazioni per la scala Mel, la più utilizzata per convertire un valore $f$ espresso in Hz in un valore $m$ in Mels è la seguente

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{4.3}$$

Di seguito il grafico della funzione Mel, un cui si può vedere l'andamento della scala Mel rispetto alla scala lineare delle frequenze. Notiamo che fino ai $1000Hz$ l'andamento è lineare, dopo l'andamento è logaritmico.

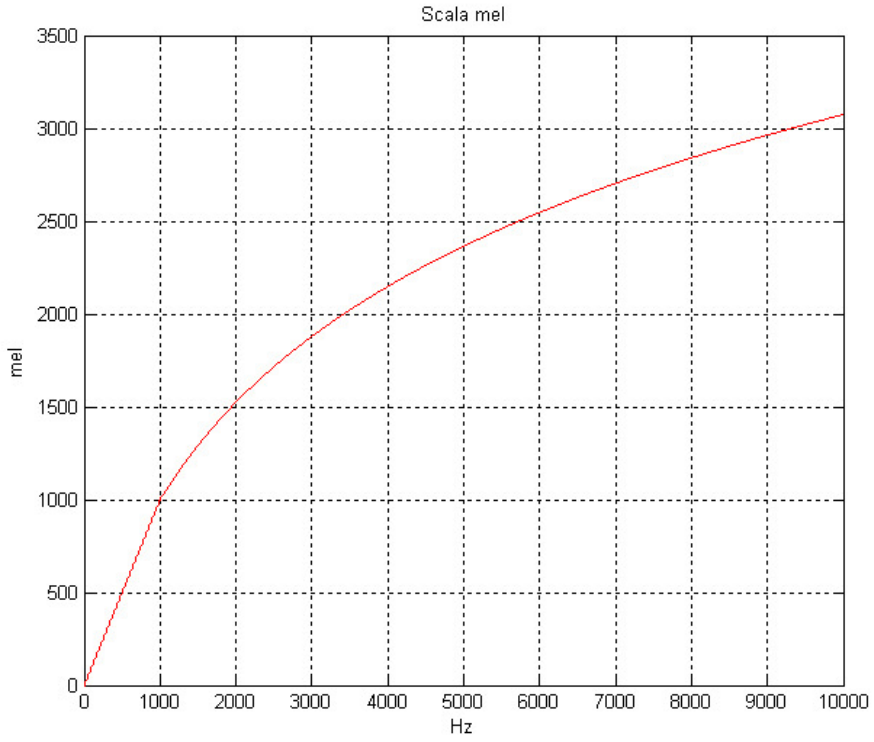



Figura 4.8: Funzione Mel.

Per calcolare lo spettrogramma di Mel, si procede inizialmente calcolando lo spettrogramma tradizionale, come descritto in precedenza. Successivamente, si applicano dei filtri passa-banda (noti come filtri di Mel, Figura 4.9) allo spettrogramma ottenuto. Questi filtri sono progettati per trasformare lo spettrogramma in uno spettrogramma di Mel, il quale rispecchia meglio la percezione umana delle frequenze. In particolare, danno maggiore risoluzione alle frequenze basse e minore risoluzione alle frequenze alte, imitando il modo in cui l'orecchio umano percepisce i suoni. Ogni filtro permette il passaggio delle frequenze comprese in un determinato intervallo e attenua le altre. La distribuzione dei filtri lungo l'asse delle frequenze non è lineare: i filtri sono più densi alle basse frequenze e più radi alle alte frequenze. Questa configurazione rispecchia il modo in cui l'orecchio umano è più sensibile alle variazioni nelle basse frequenze rispetto a quelle alte. I filtri di Mel sono tipicamente triangolari, ciascuno con un centro, una larghezza e un'altezza che determinano la gamma di frequenze che coprono e l'attenuazione che applicano. Infine, i valori ottenuti vengono trasformati in scala logaritmica.

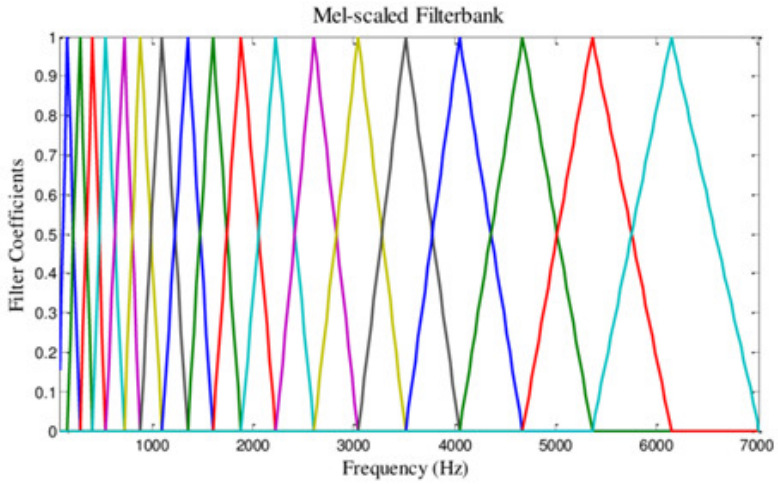



Figura 4.9: Filtri di Mel. [22]

A seguire, presentiamo un esempio di spettrogramma Mel, che illustra la rappresentazione visiva delle frequenze sonore nel tempo.

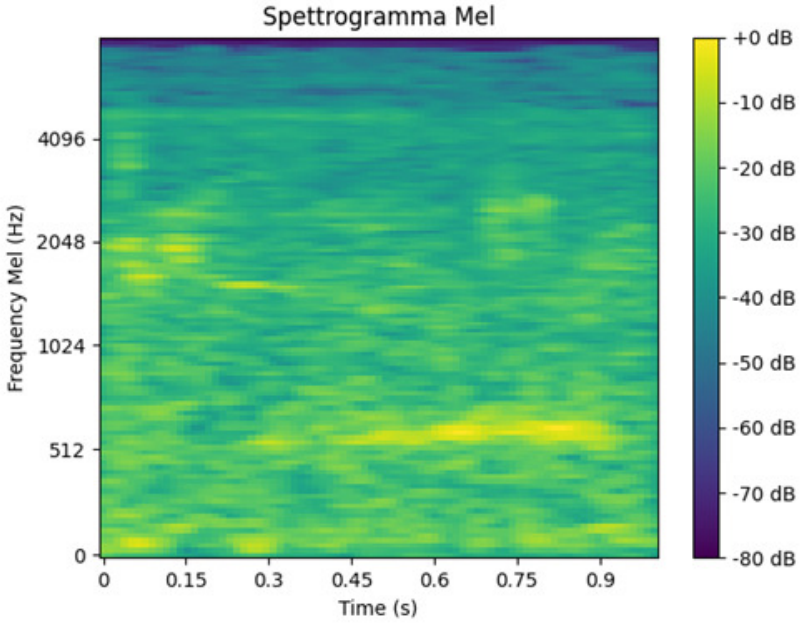



Figura 4.10: Spettrogramma Mel.

## 4.5 Tecniche di riduzione del rumore

Per migliorare la qualità degli audio grezzi e ridurre i rumori presenti, vengono applicati diversi filtri di pre-elaborazione. Questi sono essenziali per migliorare i risultati delle reti neurali che verranno addestrate con questi dati. Dopo aver applicato questi filtri, i dati audio vengono trasformati in immagini calcolando i relativi spettrogrammi di Mel, che vengono poi utilizzati per l'addestramento delle reti. I filtri che sono stati utilizzati in questo progetto sono:

- Noisereduce.
- Filtro passa alto.
- Filtro mediano.

### 4.5.1 Filtro "Noisereduce"

Noisereduce [1] è un algoritmo di riduzione del rumore in Python che riduce il rumore nei segnali nel dominio del tempo, come il parlato. Si basa su un metodo chiamato "spectral gating", che consiste nell'applicare una soglia variabile (o "gate") alle componenti di frequenza di un segnale audio per attenuare o eliminare il rumore presente. Questo metodo funziona nel seguente modo:

1. Calcolo dello spettrogramma, il primo passo è calcolare lo spettrogramma del segnale audio.
2. Stima della soglia di rumore, viene stimata una soglia di rumore per ciascuna banda di frequenza dello spettrogramma. Questa soglia rappresenta il livello al di sotto del quale i componenti del segnale sono considerati rumore.
3. Calcolo della maschera di Gating, viene quindi calcolata una maschera che permette di "bloccare" i componenti di frequenza che si trovano al di sotto della soglia di rumore. In altre parole, i componenti che sono più deboli della soglia vengono attenuati o eliminati, mentre quelli che sono più forti della soglia vengono mantenuti.

[1] https://pypi.org/project/noisereduce/

4. Applicazione della maschera, la maschera viene applicata allo spettrogramma del segnale originale, riducendo così l'ampiezza delle frequenze considerate rumore.

5. Inversione dello spettrogramma, lo spettrogramma filtrato viene convertito nuovamente nel dominio del tempo, ottenendo così un segnale audio con meno rumore.

Questo algoritmo può essere utilizzato sia nel caso di rumori stazionari che non stazionari. Nel caso di rumori stazionari, viene mantenuta la soglia di rumore stimata allo stesso livello su tutto il segnale. Mentre in quello non stazionario, la soglia di rumore stimata viene aggiornata continuamente nel tempo. Nel presente studio si è utilizzato la riduzione del rumore non stazionario.

### 4.5.2 Filtro passa alto

Un filtro passa-alto di Butterworth è un tipo di filtro progettato per permettere il passaggio delle frequenze superiori ad una certa frequenza di taglio, attenuando al contempo le frequenze che risultano inferiori a tale soglia. Questo tipo di filtro è noto per la sua risposta in frequenza estremamente piatta nella banda passante, il che significa che non introduce ondulazioni o distorsioni significative nella gamma di frequenze che lascia passare. La transizione tra la banda passante e la banda di stop è determinata dall'ordine del filtro, più alto è l'ordine più ripida è la transizione. L'attenuazione del segnale nelle frequenze inferiori alla frequenza di taglio aumenta con l'ordine del filtro. Per esempio, un filtro del primo ordine ha un'attenuazione pari a $20dB$ per decade, uno di secondo ordine ha un'attenuazione di $40dB$ per decade, uno di terzo di $60dB$ per decade e così via.
La risposta in frequenza di un filtro di ordine $n$ può essere definita matematicamente come:

$$|G(j\omega)| = \frac{1}{\sqrt{1+\omega^{2n}}} \tag{4.4}$$

dove $G$ è il trasferimento del filtro, $n$ è l'ordine del filtro, $\omega$ è il rapporto tra la frequenza del segnale e la frequenza di taglio che si vuole imporre con il filtro.

### 4.5.3 Filtro mediano

Il filtro mediano è un tipo di filtro non lineare utilizzato per ridurre il rumore nei dati audio e nelle immagini. Il suo funzionamento si basa sul calcolo del valore mediano all'interno di una finestra mobile di dimensione predefinita che scorre lungo il segnale. Per ogni posizione della finestra che scorre attraverso il segnale audio:

- I campioni presenti nella finestra vengono ordinati in ordine crescente.
- Viene selezionato il valore mediano, ovvero il valore centrale nell'insieme ordinato di campioni all'interno della finestra.
- Questo valore mediano sostituisce il valore centrale originale della finestra.

Mentre nel caso di filtri lineari il valore centrale della finestra viene sostituito con la media dei valori presenti all'interno di questo intervallo, nel caso non lineare si utilizza il valore mediano, il che permette di preservare meglio i bordi nei dati e rimuovere efficacemente il rumore impulsivo, come i picchi di rumore che possono essere presenti nei segnali audio. Con bordo di un segnale si fa riferimento ad una regione in cui sono presenti variazione rapide o brusche nei valori del segnale, come cambiamenti rapidi nell'ampiezza o nella frequenza. Preservare i bordi risulta importante, poiché possono contenere eventi significativi o caratteristiche distintive del segnale. La scelta della dimensione della finestra (o kernel) è molto importante. Una finestra troppo piccola potrebbe non rimuovere efficacemente il rumore, mentre una troppo grande potrebbe distorcere il segnale originale. Tipicamente, per segnali audio si utilizza una finestra che varia dai 3 ai 7 campioni. Questo filtro risulta computazionalmente oneroso, specialmente su grandi dataset, poiché richiede l'ordinamento dei valori all'interno di ogni finestra.

## 4.6 Metriche

Nel contesto dell'analisi e classificazione degli audio, è fondamentale disporre di strumenti che permettano di valutare accuratamente le prestazioni dei modelli di apprendimento automatico utilizzati. Le metriche giocano un ruolo cruciale nel processo, poiché consentono di quantificare quanto efficacemente i modelli, come le reti neurali convoluzionali (CNN), riescano a distinguere correttamente tra le diverse classi di dati. Essi ci aiutano a comprendere non solo il numero di previsioni corrette effettuate, ma anche la capacità del modello di bilanciare l'identificazione dei veri positivi rispetto ai falsi positivi e falsi negativi. Per misurare e valutare in maniera precisa le prestazioni di un modello di apprendimento automatico, è necessario fare uso di indicatori specifici, che consentono di determinare quanto bene il modello stia svolgendo il compito per cui è stato progettato. Prima di esplorare le metriche utilizzate per valutare i modelli, è importante definire alcuni concetti fondamentali:

- Veri negativi (TN): gli esempi che sono stati correttamente classificati come negativi. In altre parole, il modello ha identificato correttamente i casi che non appartengono alla classe di interesse.

- Veri positivi (TP): gli esempi che sono stati correttamente classificati come positivi. Questi sono i casi in cui il modello ha identificato correttamente i dati che appartengono alla classe di interesse.

- Falsi negativi (FN): gli esempi che sono stati erroneamente classificati come negativi, ma che in realtà sono positivi. Questo indica che il modello ha mancato di identificare alcuni casi rilevanti.

- Falsi positivi (FP): gli esempi che sono stati erroneamente classificati come positivi, mentre in realtà sono negativi. Questo mostra che il modello ha identificato erroneamente alcuni casi come appartenenti alla classe di interesse.

Con questi concetti in mente, possiamo descrivere le metriche principali utilizzate per valutare le prestazioni dei modelli:

- Accuratezza: si intende la percentuale di predizioni corrette rispetto al numero totale di predizioni effettuate. Fornisce una misura generale dell'efficacia del modello. La formula per l'accuratezza è la seguente:
$$accuratezza = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

- Precisione: esprime il tasso di corretta classificazione delle solo istanze positive. Si calcola come:
$$precisione = \frac{TP}{TP + FP} \tag{4.6}$$
Esempio: Immagina un sistema di allarme con riconoscimento facciale che notifica quando entra qualcuno non riconosciuto. Un modello con alta precisione avviserà raramente senza motivo, e quando lo fa, possiamo essere abbastanza certi che si tratti di un intruso. Questo significa che il modello è molto preciso nel distinguere tra intrusi e familiari.

- Recall (Richiamo): esprime il tasso di corretta classificazione delle istanze positive rispetto a tutte le istanze che erano effettivamente positive. Si esprime come:
$$recall = \frac{TP}{TP + FN} \tag{4.7}$$
Esempio: Se sei un radiologo che utilizza un modello di computer vision per rilevare tumori ai polmoni, desideri un alto recall. Questo perché è cruciale identificare tutti i tumori presenti, anche se il modello potrebbe segnalare falsi positivi. Un modello con alto richiamo ridurrà il rischio che un tumore non venga rilevato, anche se questo significa che potrebbero esserci alcuni falsi allarmi che dovranno essere esaminati. Mentre la precisione si concentra sulla qualità delle previsioni positive (cioè, quanti dei casi previsti come positivi sono effettivamente positivi), il richiamo si concentra sulla capacità del modello di trovare tutti i casi positivi presenti nei dati. A seconda del contesto e delle priorità, può essere più importante ottimizzare l'una o l'altra metrica.

- F1-Score: è una metrica che combina precisione e richiamo in un'unica misura. La formula per calcolare l'F1-Score è:
$$F1 - Score = 2 \times \frac{Precisione \times Richiamo}{Precisione + Richiamo} \tag{4.8}$$

- MSE (Mean Square Error): l'errore quadratico medio è una metrica utilizzata per valutare la discrepanza tra i valori predetti da un modello e i valori reali osservati. In altre parole, l'MSE misura la quantità di errore presente nelle predizioni del modello. È definita come:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.9}$$

  dove:

  - $n$ è il numero totale di osservazioni.
  - $y_i$ è il valore reale.
  - $\hat{y}_i$ è il valore previsto dal modello.

# Capitolo 5

# Test condotti e risultati conseguiti

In questo capitolo verranno presentati una descrizione dettagliata del dataset utilizzato e i test condotti durante lo studio. La trattazione inizierà con un'analisi delle caratteristiche principali del dataset, seguita da una descrizione della metodologia di addestramento dei modelli, con particolare attenzione alle tecniche e ai parametri impiegati durante il processo. Infine, discuteremo i test effettuati e analizzeremo i risultati ottenuti, fornendo un quadro chiaro delle performance dei modelli.

## 5.1 Dataset utilizzato

L'obbiettivo di questa ricerca è identificare con buona accuratezza quando una donna o un bambino si trovano in situazione di pericolo. Esistono numerosi dataset per la classificazione di situazioni di pericolo, ma pochi si riferiscono specificamente al contesto di questo studio. Dopo numerose ricerche, è stato individuato un solo dataset che rispondeva ai requisiti del seguente studio: il dataset pubblico del paper "Audio signal based danger detection using signal processing and deep learning" [1]. Questo dataset è composto da tre classi: bambini in pericolo, donne in pericolo e situazioni normali. Gli audio dei bambini provengono da un'organizzazione situata in Bangladesh, mentre le registrazioni delle urla delle donne sono state estratte da film e serie horror. Le situazioni normali includono registrazioni di attività quotidiane. Per aumentare il numero di dati a disposizione, i ricercatori hanno eseguito

[1] https://data.mendeley.com/datasets/gfvsdtnf3v/1

un'operazione di data augmentation (aumento dei dati). Questa tecnica è utilizzata nell'apprendimento automatico per aumentare la quantità di dati a disposizione generando nuove varianti a partire dai dati esistenti. La data augmentation risulta particolarmente utile in settori come il riconoscimento vocale, la sintesi vocale e la classificazione delle immagini, dove ottenere grandi quantità di dati etichettati può essere costoso e dispendioso in termini di tempo. Le tecniche più comuni di data augmentation includono:

- Scaling: consiste nel variare l'ampiezza del segnale audio senza modificarne le caratteristiche come la frequenza o la durata. In altre parole, cambia il volume del segnale, rendendolo più forte o più debole, senza alterare il contenuto temporale o le frequenze del segnale.
- Pitch Shifting: va a modificare l'altezza (pitch) di un suono senza alterarne la durata. Consente di cambiare la tonalità di una registrazione audio, rendendola più alta o più bassa.

Aumentando la quantità di dati a disposizione si riduce la possibilità che il nostro modello sia soggetto ad overfitting. Dopo l'operazione di data augmentation, il dataset contiene un totale di 19.375 file audio campionati con un sample rate di 40.000, ognuno della durata di 1 secondo, strutturati nel seguente modo:

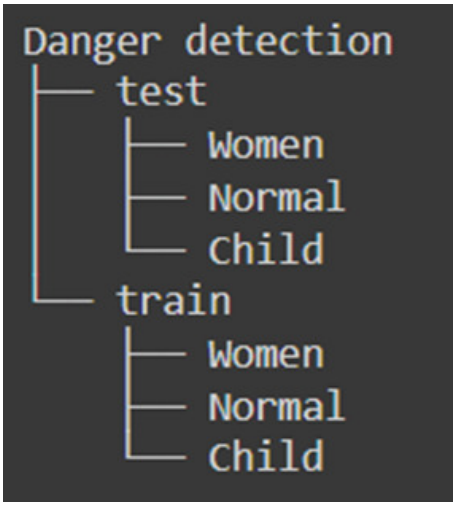



Figura 5.1: Struttura del dataset.

Come si può osservare gli audio sono suddivisi in tre classi: Child (bambini in pericolo), Normal (situazioni in cui non c'è pericolo) e Women (donne in pericolo). A loro volta, sono suddivisi in dati di addestramento (train) e dati di test con un rapporto di circa 70:30 (Figura 5.1). Di seguito, una tabella che riassume, in maniera dettagliata, il numero di audio e la loro suddivisione all'interno del dataset:

| **Classe** | **Train** | **Test** | **Totale** |
|---|---|---|---|
| Normal | 4522 | 1938 | 6460 |
| Women | 4498 | 1929 | 6427 |
| Child | 4542 | 1946 | 6488 |
| Totale | 13562 | 5813 | 19375 |

Tabella 5.1: Tabella in cui si indica la distribuzione degli audio all'interno del dataset.

L'utilizzo dell'intero dataset ha presentato alcuni problemi. In primo luogo, l'addestramento utilizzando solo una CPU comportava tempi estremamente lunghi, rendendo impossibile completare l'addestramento in un tempo ragionevole. Per questo motivo, si è deciso di utilizzare una GPU messa a disposizione da Google Colab. Tuttavia, Google Colab impone limitazioni sull'uso delle sue GPU, permettendo di utilizzarle per un determinato periodo di tempo. Una volta scaduto questo tempo, il sistema passa automaticamente all'uso della CPU, prolungando i tempi di elaborazione in modo significativo. Per ovviare a queste limitazioni e riuscire ad addestrare e testare i modelli in tempi ragionevoli, si è deciso di lavorare su un dataset ridotto.
Il dataset ridotto utilizzato per l'addestramento consiste in 1000 audio per ciascuna delle tre classi, per un totale di 3000 audio. Per i test, è stato impiegato un campione di 500 audio per classe, per un totale di 1500 audio. Ogni classe è stata rappresentata con un numero uguale di file sia per l'addestramento che per i test. Questo bilanciamento delle classi assicura che il modello riceva una rappresentazione equa di ciascuna classe durante l'addestramento, migliorando la sua capacità di distinguere in modo accurato e affidabile tra le diverse classi. Questa strategia ha permesso di ottimizzare i tempi senza compromettere eccessivamente le prestazioni del modello.

## 5.2 Addestramento modelli

Prima di addestrare i modelli, come prima operazione, ho applicato le tecniche di riduzione del rumore precedentemente descritte al dataset, in modo da avere a disposizione per l'addestramento e i test audio già ripuliti. Per ogni dataset "ripulito" dai rumori ("Dataset_originale", "Dataset_noisereduce", "Dataset_highpass_filter", e "Dataset_median_filter") ho poi applicato lo stesso procedimento per addestrare e testare le reti.
I dati di addestramento sono stati suddivisi in due insiemi principali. Il primo insieme, denominato wav_train, contiene i file audio destinati all'addestramento del modello, mentre il secondo insieme, denominato wav_val, è stato creato estraendo il 10% dei dati originariamente destinati all'addestramento. Questo secondo insieme viene utilizzato per la validazione del modello, permettendo di monitorare le sue prestazioni su dati non utilizzati durante l'addestramento. A ciascuno di questi insiemi sono associati i rispettivi vettori di etichette (label_train per wav_train e label_val per wav_val), che rappresentano le classi di appartenenza dei file audio. Per calcolare le etichette, ai nomi delle classi è stato associato un indice numerico: l'indice 0 per la classe Child, l'indice 1 per la classe Normal e l'indice 2 per la classe Women. Questi indici consentono al modello di comprendere a quale classe appartiene ciascun file audio durante l'addestramento e la validazione.
Prima di approfondire il processo di addestramento, è utile chiarire due concetti fondamentali: batch ed epoche. Un batch è un sottoinsieme del set di dati che viene utilizzato per aggiornare i pesi del modello durante l'addestramento. Quando un modello viene addestrato, i dati vengono suddivisi in piccoli gruppi chiamati batch. Il modello elabora ogni batch separatamente, calcola la perdita (ossia quanto il modello si è discostato dalla previsione corretta) e poi utilizza questa informazione per aggiornare i pesi della rete. Un'epoca è un ciclo completo in cui il modello vede l'intero set di dati di addestramento una volta. In altre parole, durante un'epoca, il modello elabora tutti i batch che compongono il set di dati.
L'addestramento è stato eseguito con un batch size di 4, suddividendo il set di dati in piccoli gruppi di 4 campioni per volta. Il processo di apprendimento è stato condotto su 50 epoche, permettendo al modello di vedere l'intero set di dati di addestramento 50 volte. Inoltre, ad ogni epoca, i batch sono stati mescolati casualmente per garantire che il modello non imparasse un ordine specifico dei dati, migliorando

così la sua capacità di generalizzare. Alla fine di ogni epoca, il modello utilizza il set di validazione (wav_val e label_val) per valutare le sue prestazioni, nello specifico il valore della loss e dell'accuratezza. Questo processo di valutazione consente di monitorare l'efficacia del modello su dati non visti durante l'addestramento. La valutazione sui dati di validazione aiuta a prevenire l'overfitting, assicurando che il modello non solo migliori le sue performance sui dati di addestramento, ma anche che sia in grado di generalizzare bene su nuovi dati.
Per l'allenamento delle nostre reti abbiamo usato i seguenti parametri:

| **Learning rate** | **Batch size** | **Epoche** | **Funzione perdita** | **Ottimizzazione** | **Metriche** |
|---|---|---|---|---|---|
| 0.001 | 4 | 50 | Categorical Cross-Entropy | Adam | Accuracy |

Tabella 5.2: Parametri delle reti.

I set di validazione e addestramento passati alle reti non contengono la rappresentazione tempo-frequenza del segnale. Per ottenere questa rappresentazione è stato aggiunto all'inizio di ogni rete una funzione della libreria Kapre (Keras audio preprocessors) [2], utilizzata per creare un livello (layer) di pre-elaborazione audio che genera uno spettrogramma di Mel a partire da un segnale audio grezzo. Questa funzione utilizza la STFT per calcolare lo spettrogramma con i seguenti parametri:

- Una FFT di 512, con cui si intende l'applicazione della fast fourier transform ad un blocco di 512 campioni del segnale.
- Una window length di 400, ovvero la finestra utilizzata per calcolare la FFT ha una lunghezza di 400 campioni.
- Un hop length (passo) di 160, è il numero di campioni di cui si sposta la finestra in avanti ogni volta che si calcola la FFT. Questo parametro determina anche la sovrapposizione delle finestre, in questo caso la sovrapposizione è di 240 campioni.

[2]https://kapre.readthedocs.io/en/latest/

Allo spettrogramma ottenuto applichiamo una serie di 128 filtri Mel per ottenere lo spettrogramma di Mel. Di seguito, presentiamo un esempio basato sugli audio del dataset, che illustra la forma d'onda e il corrispondente spettrogramma Mel, ottenuto tramite il layer di pre-elaborazione posto all'inizio di ciascuna rete.

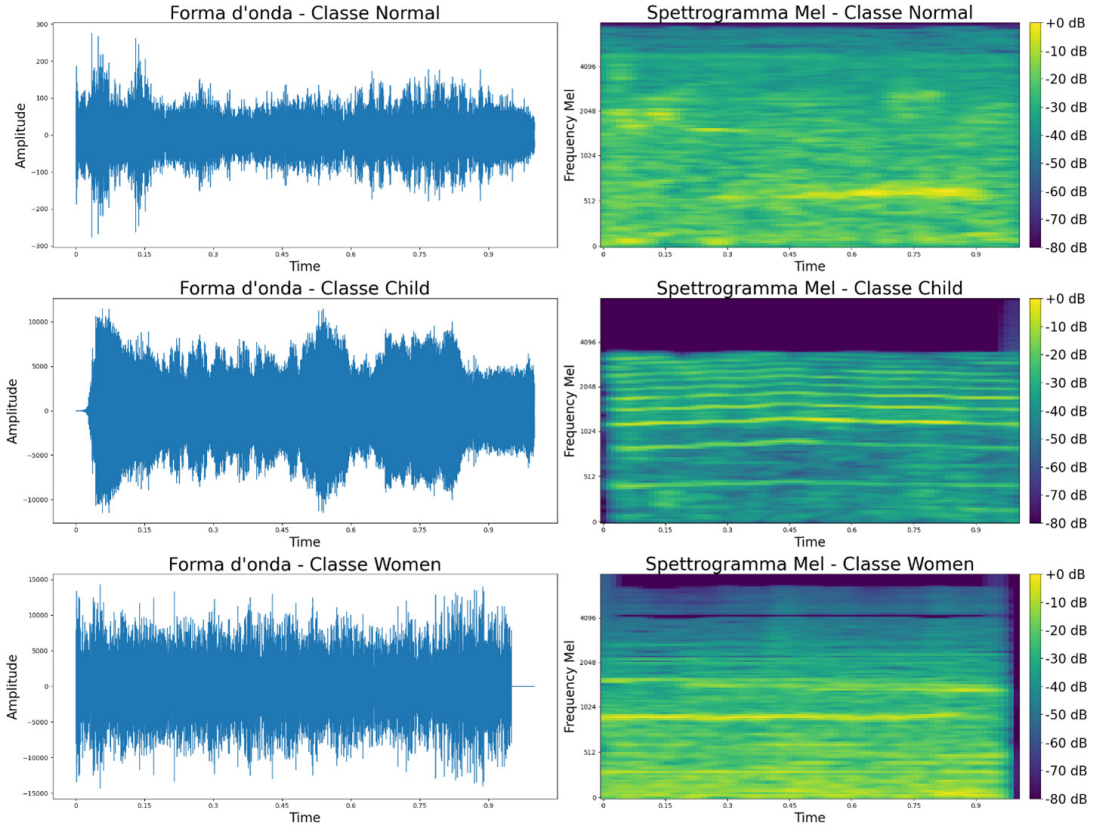



Figura 5.2: A sinistra sono visualizzate le forme d'onda per ciascuna classe, mentre a destra sono riportati gli spettrogrammi Mel corrispondenti. Questa figura illustra le differenze tra le caratteristiche temporali e frequenziali dei segnali associati alle classi "Normal", "Child" e "Women".

Questi spettrogrammi (Figura 5.2, grafici a destra) vengono utilizzati per l'addestramento delle reti. Poiché i dati disponibili non sono sufficientemente grandi per addestrare una rete da zero, adottiamo la tecnica del transfer learning sui nostri modelli. Questa strategia sfrutta le competenze che una rete ha sviluppato su un ampio dataset etichettato, applicandole a un nuovo compito che dispone di un numero ridotto di dati. I nostri modelli sono stati pre-addestrati su ImageNet, un vasto database contenente milioni di immagini etichettate in migliaia di categorie.

In questo modo, la rete è in grado di riconoscere numerose caratteristiche visive, come bordi, texture, forme e colori. L'uso del transfer learning offre diversi vantaggi: una maggiore accuratezza anche con un numero limitato di dati, un notevole risparmio di tempo e risorse rispetto all'addestramento da zero, che richiederebbe molto più tempo e potenza di calcolo, e la possibilità di applicarlo a piccoli dataset, sfruttando la solida base di conoscenze che la rete ha già acquisito grazie a un grande dataset.
Durante il processo di addestramento, abbiamo implementato una funzione di callback specifica, progettata per monitorare l'accuratezza del modello sui dati di validazione. Questa funzione di callback è stata configurata per salvare automaticamente il modello ogni volta che l'accuratezza di validazione raggiunge un nuovo valore massimo. In questo modo, si garantisce che venga conservata la versione del modello che ha ottenuto le migliori prestazioni sui dati di validazione, evitando di sovrascriverlo con versioni meno accurate o che potrebbero soffrire di overfitting.

## 5.3 Valutazione dei Modelli: Test e Risultati

Una volta addestrate le reti, ho eseguito la fase di test, per valutare la capacità delle reti di riconoscere e classificare gli audio in una delle tre classi, utilizzando file che il modello non aveva mai visto prima. Durante la fase di previsione, passiamo al sistema gli audio presenti nella cartella test. Per ogni audio, la rete genera una probabilità associata a ciascuna classe, indicando la probabilità che l'audio appartenga a ciascuna categoria. Identifichiamo quindi la classe stimata selezionando quella con la probabilità più alta e confrontiamo questa previsione con l'etichetta corretta, per verificare l'accuratezza del modello. Per valutare la capacità di generalizzazione e robustezza dei diversi modelli di rete neurale, sono stati eseguiti test utilizzando sia il dataset originale sia vari dataset derivati da esso. Nello specifico:

1. Dataset Originale: questo dataset contiene audio non pre-processati, ossia senza l'applicazione di filtri o tecniche di riduzione del rumore. Rappresenta il riferimento principale per misurare le prestazioni dei modelli in condizioni naturali e non alterate.

2. Dataset con Riduzione del Rumore: a partire dal dataset originale, sono stati generati diversi dataset pre-processati utilizzando tecniche di riduzione

del rumore, tra cui NoiseReduce, il Filtro Passa Alto e il Filtro Mediano. Questi dataset consentono di valutare l'impatto della riduzione del rumore sulla capacità dei modelli di classificare correttamente gli audio, migliorando la robustezza dei modelli in presenza di segnali rumorosi.

I risultati ottenuti dai test sono riassunti nelle tabelle che seguono. Per ciascun modello, presentiamo due tabelle: la prima riporta l'accuratezza del modello applicando diversi filtri, insieme all'audio originale. In essa, sono evidenziate le accuratezze per le categorie 'Normal', 'Child' e 'Women', oltre all'accuratezza complessiva del modello.
La seconda tabella riassume le metriche di ciascuna rete, anche in questo caso distinguendo tra i vari filtri e l'audio originale.
In aggiunta, per una valutazione più approfondita delle previsioni del modello, oltre alle metriche e all'accuratezza, abbiamo utilizzato la matrice di confusione. Questo strumento visivo permette di analizzare la distribuzione delle previsioni corrette e degli errori commessi dal modello, fornendo una panoramica completa sulle vere previsioni positive, false previsioni negative e altre combinazioni di classificazione. Nelle matrici riportate, è possibile osservare le prestazioni di ciascun modello sul set di test: le righe rappresentano le classi reali, mentre le colonne indicano le classi predette dal modello. In questo modo, si fornisce un quadro esaustivo sull'efficacia dei singoli sistemi di classificazione.

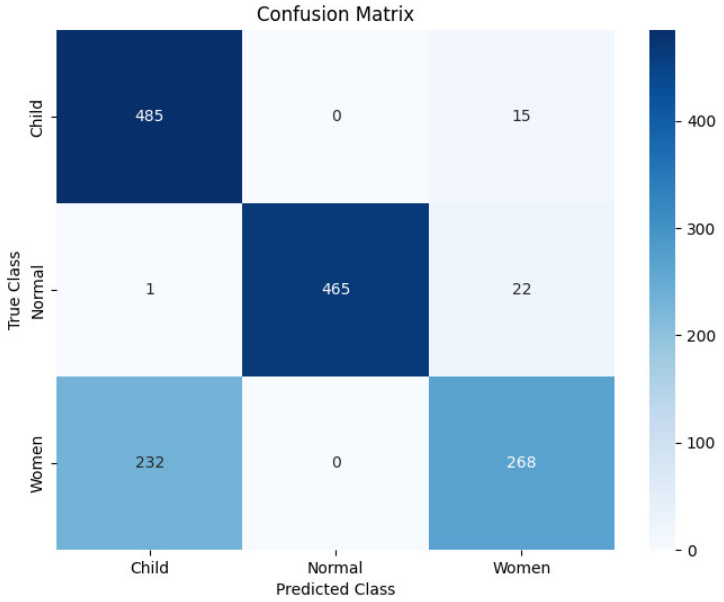
### 5.3.1 Risultati ResNet50

| Modello | Filtri | Totale | Normal | Child | Women |
|---|---|---|---|---|---|
| **ResNet50** | Audio originali | 81,86% | 95,29% | 97% | 53,6% |
| | Noisereduce | 87,1% | 96,11% | 89,8% | 75,6% |
| | Filtro passa alto | 87,3% | 97,54% | 94,2% | 70,4% |
| | Filtro mediano | 86,22% | 98,16% | 93,6% | 67,2% |

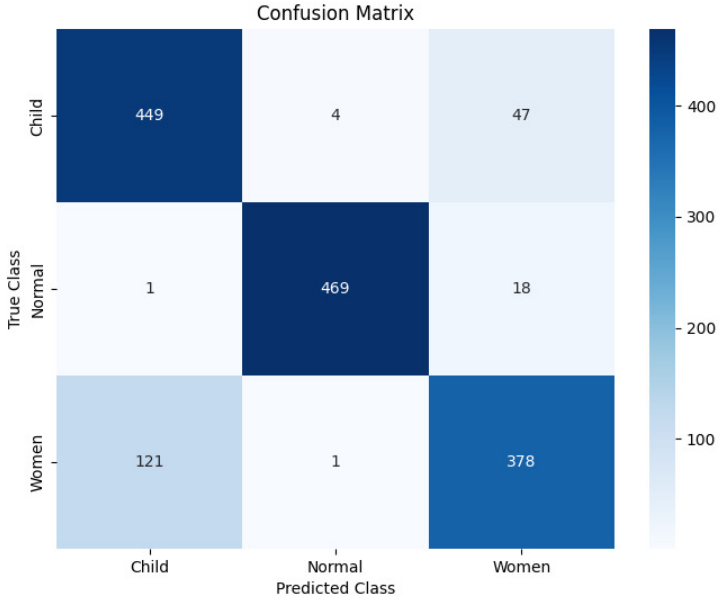
Tabella 5.3: Accuratezze ResNet50.

| Modello | Filtri | Precisione | Recall | F1 | MSE |
|---|---|---|---|---|---|
| **ResNet50** | Audio originali | 0,85 | 0,82 | 0,81 | 0,68 |
| | Noisereduce | 0,88 | 0,87 | 0,87 | 0,47 |
| | Filtro passa alto | 0,89 | 0,87 | 0,87 | 0,48 |
| | Filtro mediano | 0,88 | 0,86 | 0,86 | 0,53 |

Tabella 5.4: Metriche ResNet50.

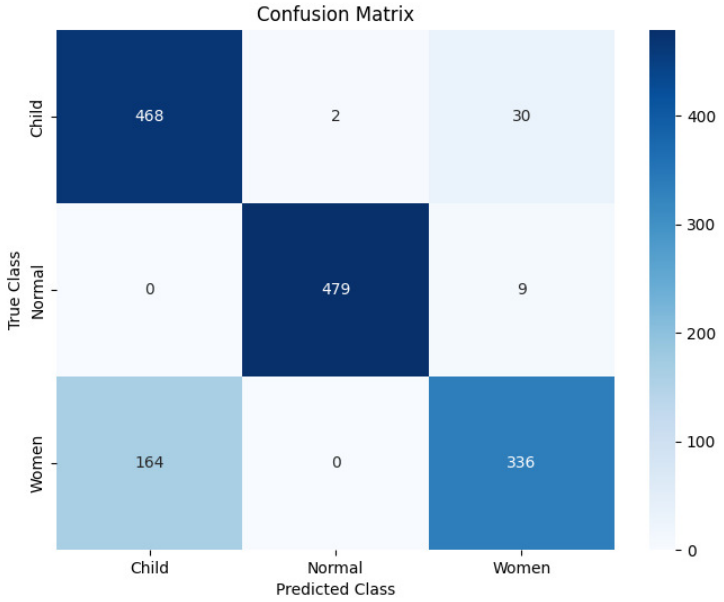



(a) Audio originali.

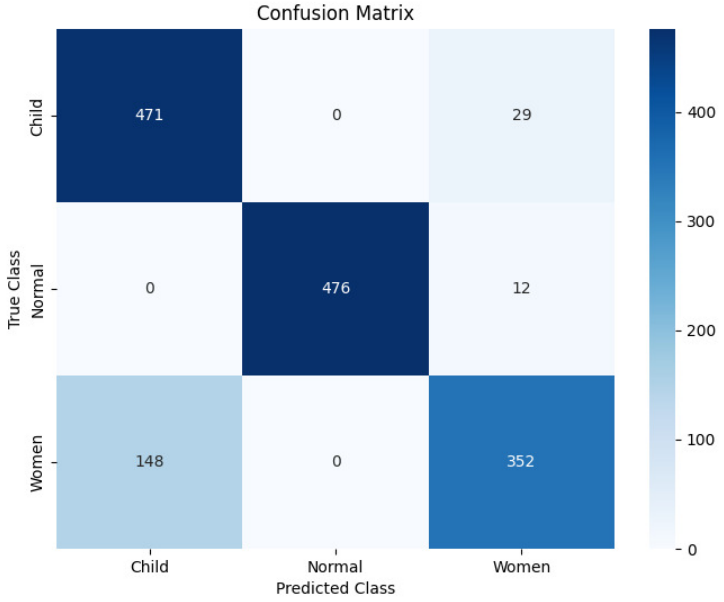



(b) Filtro noisereduce.

Figura 5.3: Matrici di confusione ResNet50.




(c) Filtro mediano.




(d) Filtro passa alto.

Figura 5.3: Matrici di confusione ResNet50.

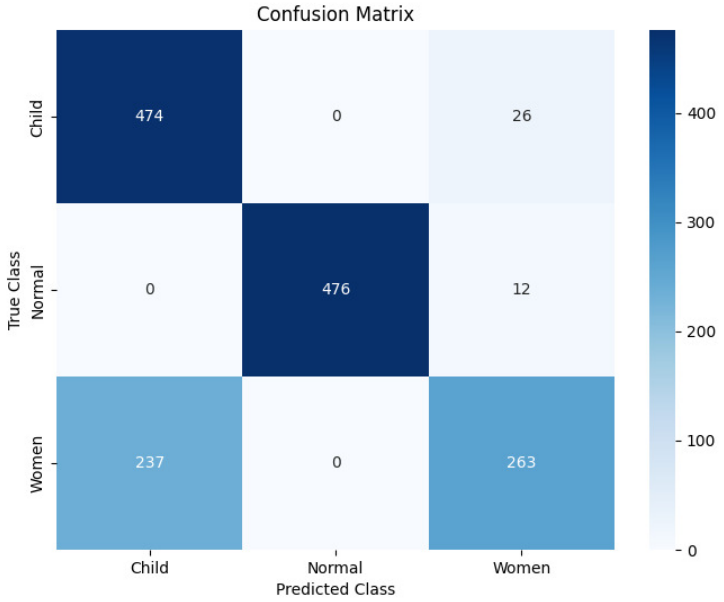
### 5.3.2 Risultati ResNet101

| Modello | Filtri | Totale | Normal | Child | Women |
|---|---|---|---|---|---|
| **ResNet101** | Audio originali | 81,52% | 97,54% | 94,8% | 52,6% |
| | Noisereduce | 88,17% | 94,67% | 86,4% | 83,6% |
| | Filtro passa alto | 89,72% | 99,39% | 92% | 78% |
| | Filtro mediano | 86,56% | 97,13% | 90% | 72,8% |

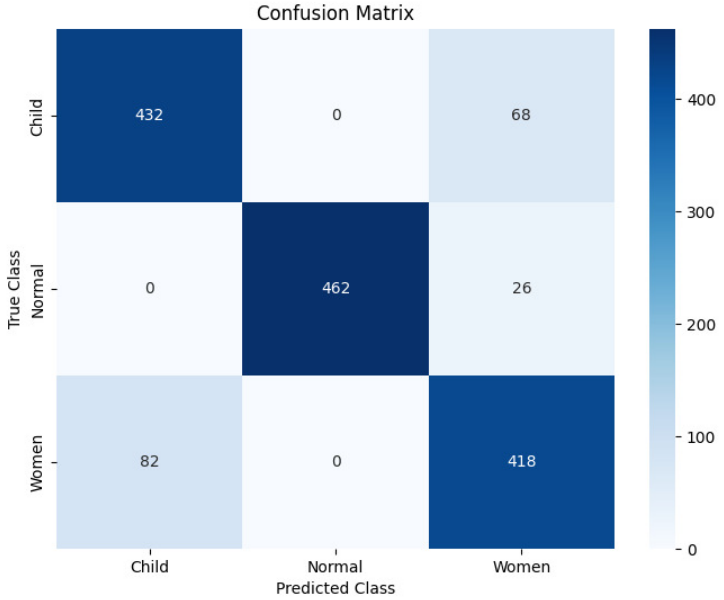
Tabella 5.5: Accuratezze ResNet101.

| Modello | Filtri | Precisione | Recall | F1 | MSE |
|---|---|---|---|---|---|
| **ResNet101** | Audio originali | 0,85 | 0,82 | 0,81 | 0,72 |
| | Noisereduce | 0,89 | 0,88 | 0,88 | 0,42 |
| | Filtro passa alto | 0,90 | 0,90 | 0,89 | 0,41 |
| | Filtro mediano | 0,87 | 0,87 | 0,87 | 0,51 |

Tabella 5.6: Metriche ResNet101.

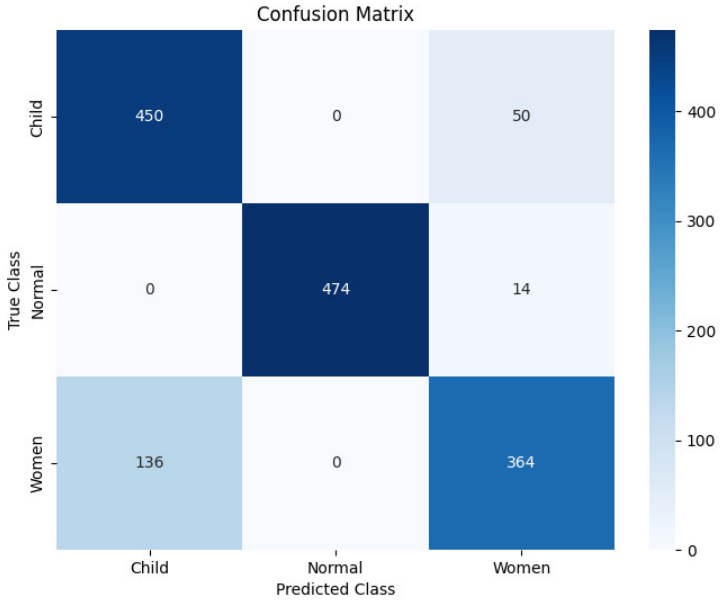



(a) Audio originali.

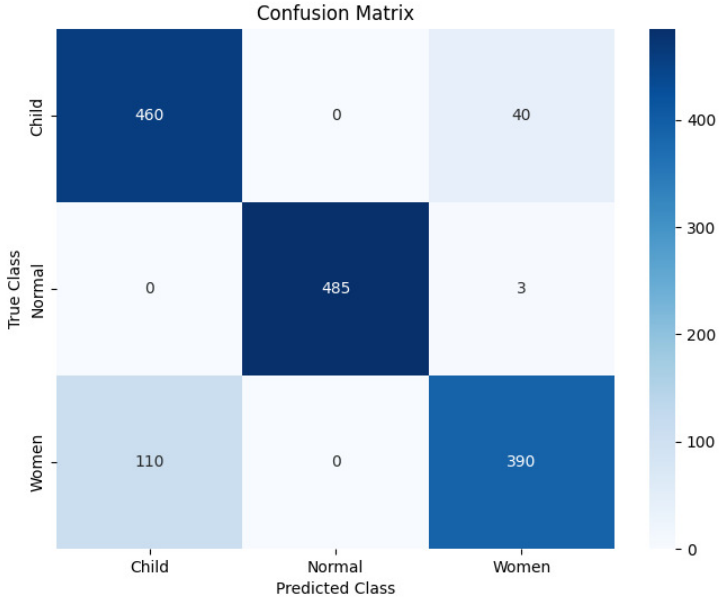



(b) Filtro noisereduce.

Figura 5.4: Matrici di confusione ResNet101.




(c) Filtro mediano.




(d) Filtro passa alto.

Figura 5.4: Matrici di confusione ResNet101.

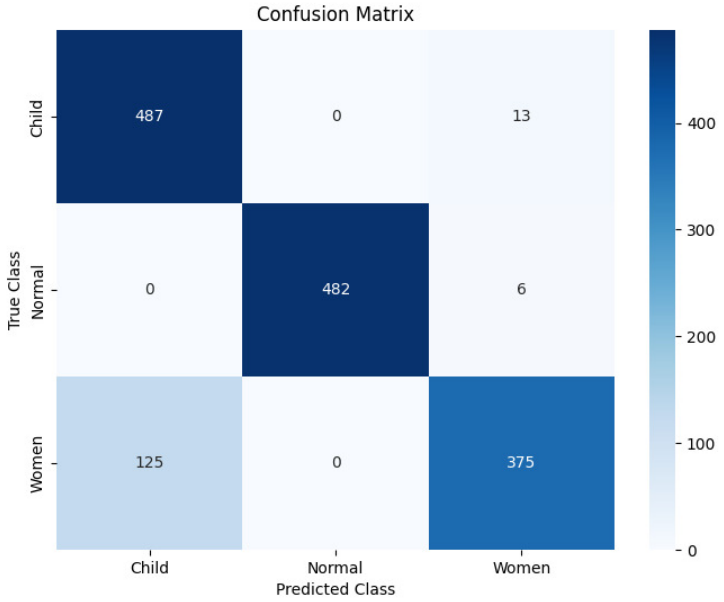
### 5.3.3 Risultati Xception

| Modello | Filtri | Totale | Normal | Child | Women |
|---|---|---|---|---|---|
| **Xception** | Audio originali | 90,32% | 98,77% | 97,4% | 75% |
| | Noisereduce | 91,26% | 99,80% | 90,2% | 84% |
| | Filtro passa alto | 87,23% | 98,98% | 94,8% | 68,2% |
| | Filtro mediano | 88,11% | 95,90% | 82,4% | 86,2% |

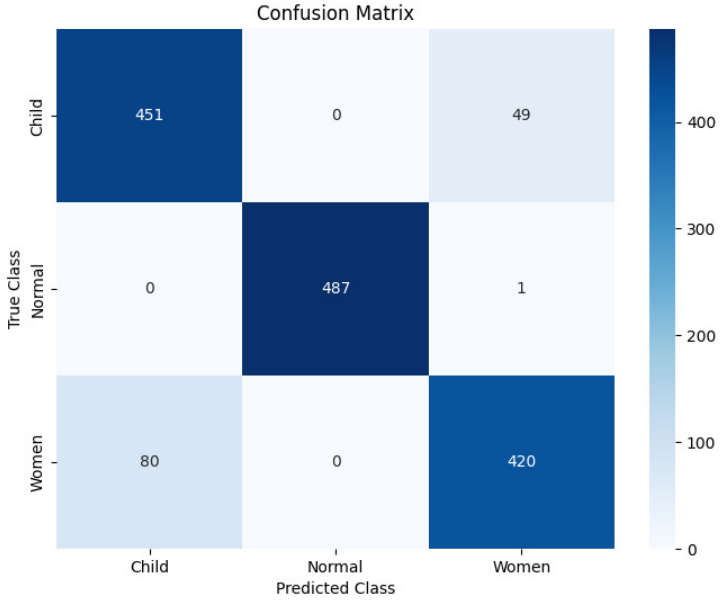
Tabella 5.7: Accuratezze Xception.

| Modello | Filtri | Precisione | Recall | F1 | MSE |
|---|---|---|---|---|---|
| **Xception** | Audio originali | 0,92 | 0,90 | 0,90 | 0,38 |
| | Noisereduce | 0,91 | 0,91 | 0,91 | 0,35 |
| | Filtro passa alto | 0,89 | 0,87 | 0,87 | 0,50 |
| | Filtro mediano | 0,89 | 0,88 | 0,88 | 0,43 |

Tabella 5.8: Metriche Xception.

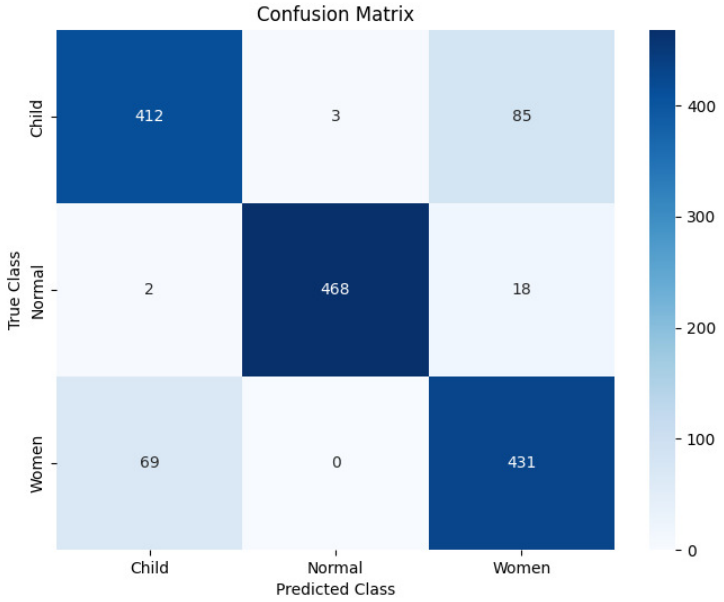



(a) Audio originali.

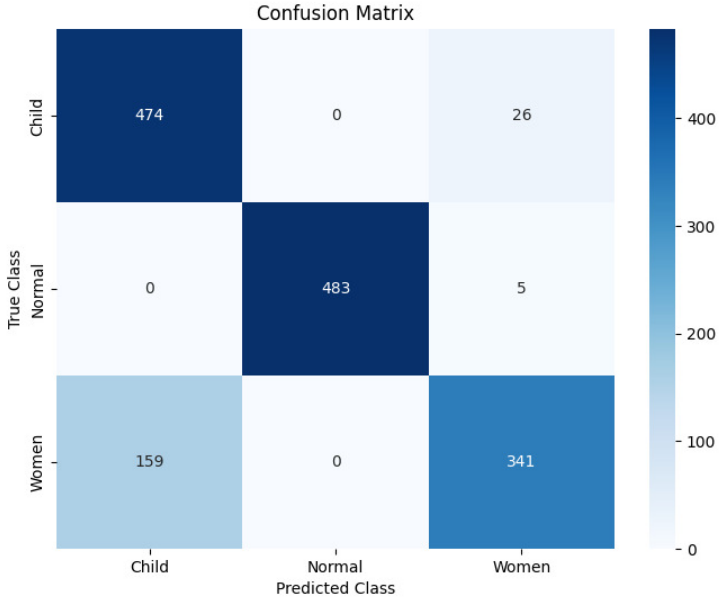



(b) Filtro noisereduce.

Figura 5.5: Matrici di confusione Xception.




(c) Filtro mediano.




(d) Filtro passa alto.

Figura 5.5: Matrici di confusione Xception.

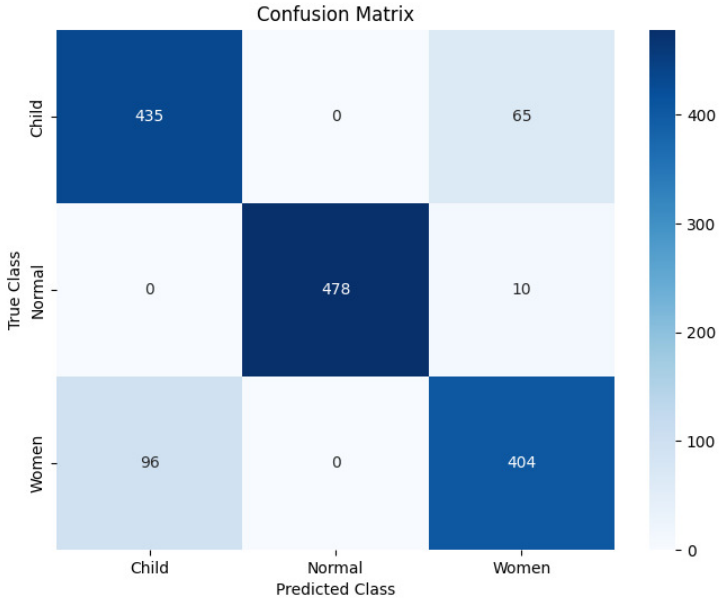
### 5.3.4 Risultati MobileNetV2

| Modello | Filtri | Totale | Normal | Child | Women |
|---|---|---|---|---|---|
| **MobileNetV2** | Audio originali | 88,51% | 97,95% | 87% | 80,8% |
| | Noisereduce | 87,63% | 98,36% | 90,6% | 74,2% |
| | Filtro passa alto | 90,59% | 99,39% | 93,2% | 79,4% |
| | Filtro mediano | 84,27% | 93,44% | 89% | 70,6% |

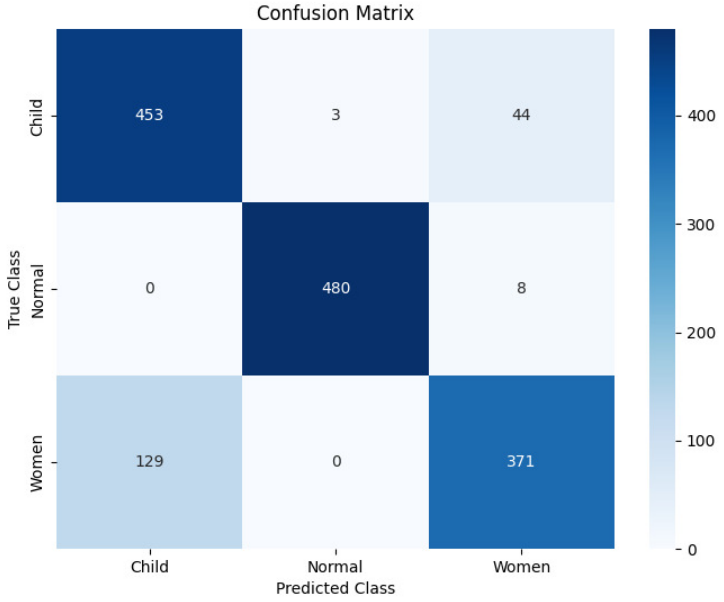
Tabella 5.9: Accuratezze MobileNetV2.

| Modello | Filtri | Precisione | Recall | F1 | MSE |
|---|---|---|---|---|---|
| **MobileNetV2** | Audio originali | 0,89 | 0,89 | 0,89 | 0,44 |
| | Noisereduce | 0,88 | 0,88 | 0,88 | 0,47 |
| | Filtro passa alto | 0,91 | 0,91 | 0,91 | 0,37 |
| | Filtro mediano | 0,85 | 0,84 | 0,84 | 0,57 |

Tabella 5.10: Metriche MobileNetV2.

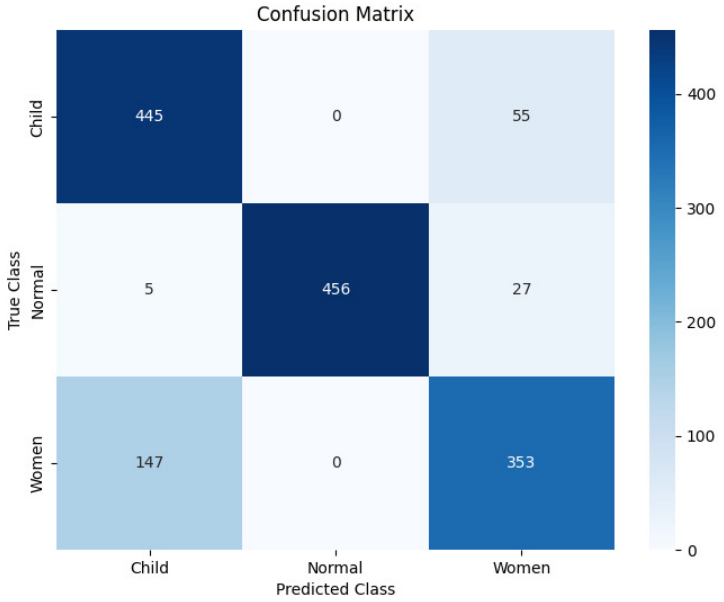



(a) Audio originali.

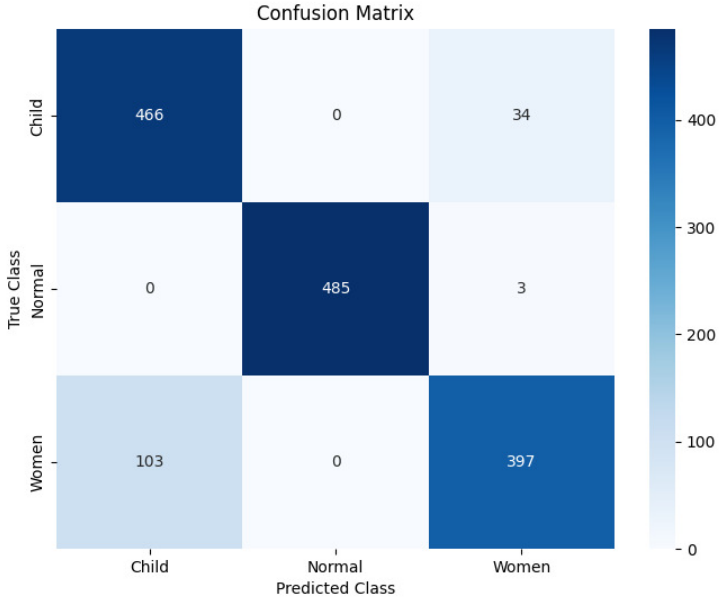



(b) Filtro noisereduce.

Figura 5.6: Matrici di confusione MobileNetV2.




(c) Filtro mediano.




(d) Filtro passa alto.

Figura 5.6: Matrici di confusione MobileNetV2.

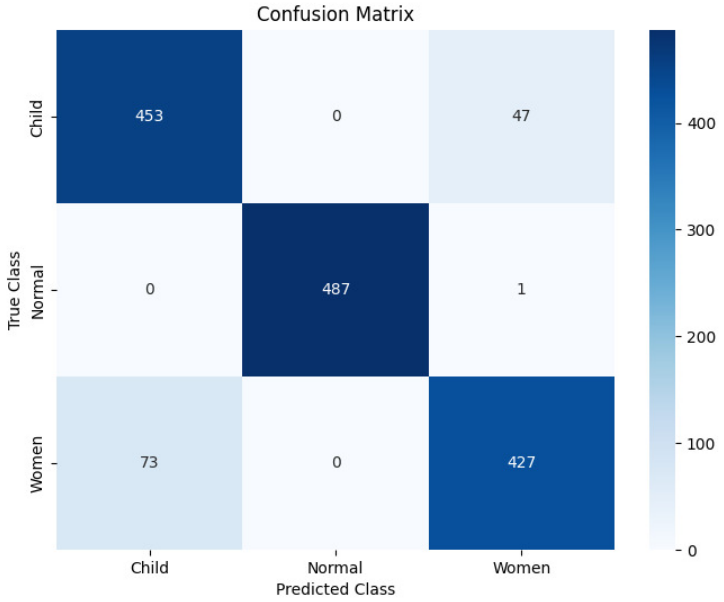
### 5.3.5 Risultati InceptionV3

| Modello | Filtri | Totale | Normal | Child | Women |
|---|---|---|---|---|---|
| **InceptionV3** | Audio originali | 91,87% | 99,80% | 90,6% | 85,4% |
| | Noisereduce | 84,07% | 96,31% | 90,6% | 65,6% |
| | Filtro passa alto | 89,79% | 99,39% | 96% | 74,2% |
| | Filtro mediano | 83,74% | 95,7% | 95,8% | 60% |

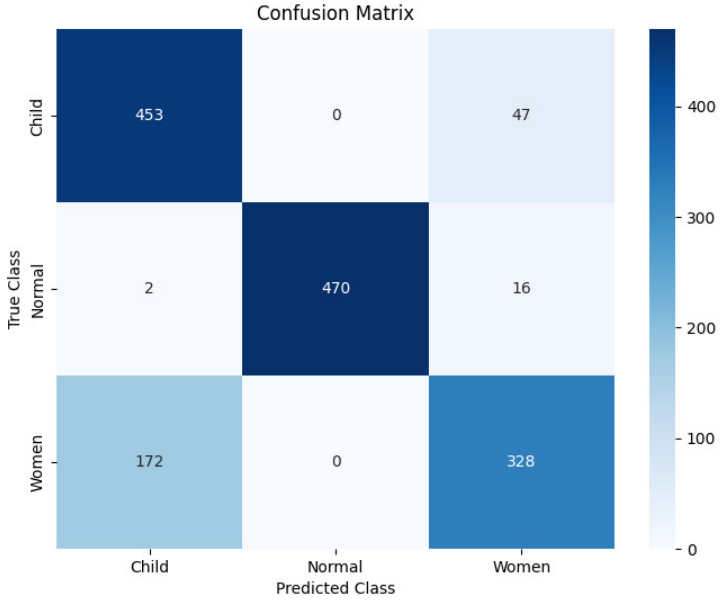
Tabella 5.11: Accuratezze InceptionV3.

| Modello | Filtri | Precisione | Recall | F1 | MSE |
|---|---|---|---|---|---|
| **InceptionV3** | Audio originali | 0,92 | 0,92 | 0,92 | 0,32 |
| | Noisereduce | 0,85 | 0,84 | 0,84 | 0,60 |
| | Filtro passa alto | 0,91 | 0,91 | 0,91 | 0,40 |
| | Filtro mediano | 0,86 | 0,84 | 0,83 | 0,60 |

Tabella 5.12: Metriche InceptionV3.

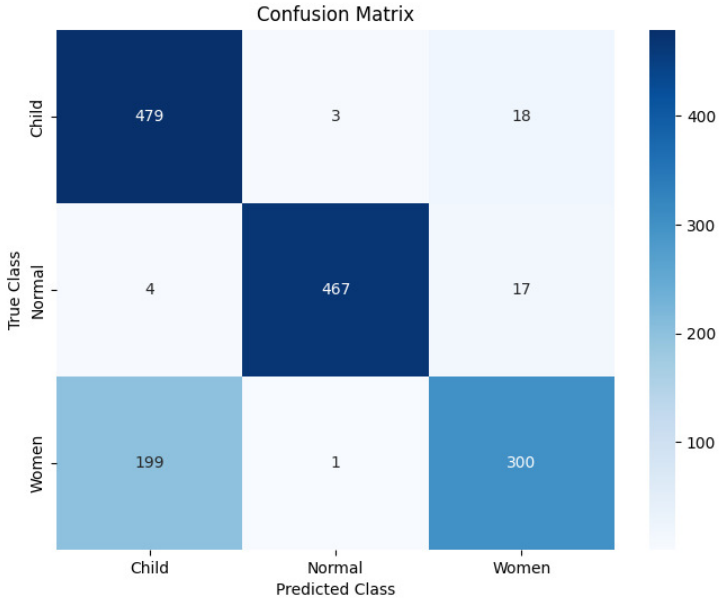



(a) Audio originali.

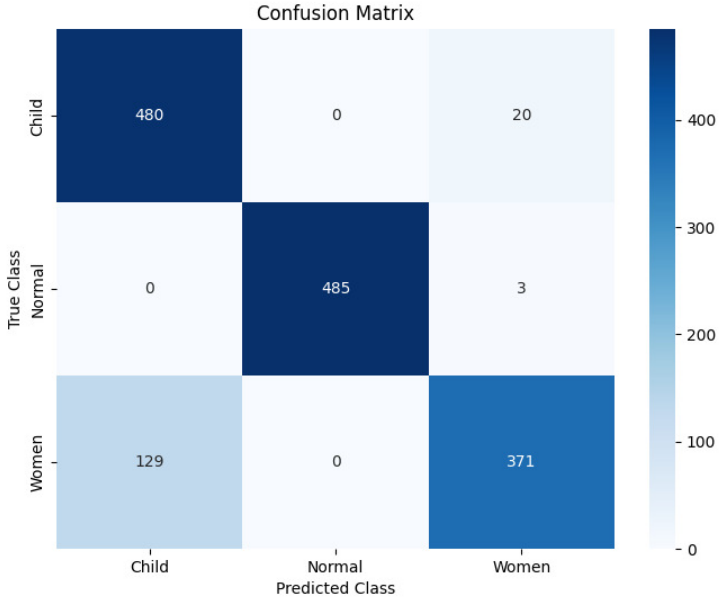



(b) Filtro noisereduce.

Figura 5.7: Matrici di confusione InceptionV3.




(c) Filtro mediano.




(d) Filtro passa alto.

Figura 5.7: Matrici di confusione InceptionV3.

## 5.4 Discussione dei risultati

Sulla base dei risultati ottenuti, si osserva che i modelli raggiungono un'accuratezza complessiva che varia dall'81% al 91%, a seconda della rete neurale e della tecnica di riduzione del rumore utilizzata. Questo dimostra la solidità dei modelli nell'identificare correttamente i segnali audio appartenenti alle diverse categorie. Tuttavia, un'analisi più approfondita delle singole classi rivela che le classi Normal e Child sono classificate con maggiore precisione rispetto alla classe Women. Ad esempio, per il modello ResNet50 utilizzato con audio originale, l'accuratezza per la classe Women è solo del 53,6%, mentre per la classe Normal è del 95,3% e per la classe Child è del 97,0%. Anche dopo l'applicazione di tecniche di riduzione del rumore, l'accuratezza della classe Women migliora ma rimane comunque inferiore rispetto alle altre due classi.
Un aspetto rilevante riguarda l'efficacia delle tecniche di riduzione del rumore. In particolare, per la rete InceptionV3 i risultati ci mostrano che l'uso degli audio originali senza applicare alcuna tecnica di riduzione del rumore ha prodotto le migliori performance. Questo contrasta con quanto osservato per le altre reti, dove il risultato migliore è stato ottenuto con l'applicazione di una delle tecniche di riduzione del rumore. Queste differenze possono essere attribuite alla struttura e all'architettura specifica di ciascuna rete neurale. Le reti come ResNet50, ResNet101, Xception, e MobileNetV2 sembrano trarre beneficio dall'applicazione di tecniche di riduzione del rumore, migliorando l'accuratezza rispetto all'utilizzo degli audio originali. Questo suggerisce che queste architetture siano particolarmente sensibili al rumore nei dati e che la riduzione del rumore riesca ad eliminare componenti indesiderate che potrebbero confondere il modello durante la fase di apprendimento e classificazione. Al contrario, nel caso della rete InceptionV3, l'applicazione di tecniche di riduzione del rumore ha portato a risultati peggiori rispetto all'utilizzo degli audio originali. Questo potrebbe essere dovuto alla capacità di InceptionV3 di gestire meglio la variabilità del rumore nei dati originali, preservando al contempo caratteristiche importanti che potrebbero essere attenuate o eliminate dalle tecniche di riduzione del rumore. L'architettura di InceptionV3, infatti, è nota per la sua capacità di catturare informazioni su scale diverse, e potrebbe quindi essere in grado di distinguere tra segnali utili e rumore in modo più efficace rispetto ad altre reti.
Le differenze nei risultati tra le varie tecniche di riduzione del rumore dipendono

anche dalle caratteristiche specifiche di ciascuna tecnica e dalla loro interazione con l'architettura della rete. Esaminando le varie tecniche di riduzione del rumore, emerge quanto segue:

- Noisereduce: Questa tecnica ha dimostrato di essere efficace in molte architetture di rete. Per alcuni modelli, riesce a ridurre il rumore mantenendo intatte le informazioni essenziali, portando a un miglioramento delle prestazioni. Tuttavia, in altre architetture, potrebbe rimuovere frequenze cruciali per la classificazione, compromettendo così l'accuratezza complessiva.

- Filtro passa-alto: In alcune reti, come ResNet50, ResNet101 e MobileNetV2, il filtro passa-alto ha portato a miglioramenti, suggerendo che l'eliminazione delle basse frequenze, spesso dominate dal rumore, consente al modello di concentrarsi su caratteristiche più informative. Tuttavia, in altre reti, l'eliminazione delle basse frequenze potrebbe aver rimosso anche informazioni utili, riducendo l'accuratezza.

- Filtro mediano: Questa tecnica ha mostrato una certa variabilità nei risultati, con alcuni modelli che hanno beneficiato della sua applicazione e altri che hanno visto un calo nelle prestazioni. Il filtro mediano è utile per eliminare i picchi di rumore impulsivo, ma può anche smussare i dettagli spettrali, compromettendo la qualità delle informazioni disponibili per il modello.

In sintesi, sebbene l'applicazione delle tecniche di riduzione del rumore abbia spesso migliorato la classificazione, l'efficacia di queste tecniche varia a seconda dell'architettura del modello e della natura dei dati audio. Questo sottolinea l'importanza di adattare le tecniche di pre-processing alle specifiche esigenze dell'architettura del modello per ottenere i migliori risultati.

# Capitolo 6

# Conclusioni e sviluppi futuri

Questo studio ha evidenziato che il riconoscimento di situazioni di pericolo, distinguendo anche se si tratta di bambini e donne, rappresenta una sfida complessa che richiede modelli avanzati e tecniche di pre-processing degli audio specifiche per ottenere risultati accurati. Ci siamo concentrati su reti neurali comunemente utilizzate nella classificazione di immagini, come ResNet50, ResNet101, Xception, MobileNetV2 e InceptionV3. I risultati ottenuti mostrano che alcuni modelli presentano buone prestazioni, mentre altri si sono rivelati meno efficaci. In particolare, siamo riusciti a raggiungere un'accuratezza pari o superiore al 90% con la rete Xception applicando l'algoritmo di riduzione del rumore noisereduce, con MobileNetV2 utilizzando il filtro passa-alto e con InceptionV3 usando i dati grezzi. Questi risultati indicano che i modelli selezionati riescono a riconoscere abbastanza bene situazioni di pericolo. Tuttavia, il problema principale emerso riguarda l'accuratezza nella classe "Women", che si è rivelata significativamente inferiore rispetto alle classi "Normal" e "Child". Questo risultato potrebbe essere dovuto alle specifiche caratteristiche spettrali delle voci femminili, oltre al fatto che alcune tecniche di riduzione del rumore potrebbero aver eliminato informazioni rilevanti per la corretta classificazione di questa classe.

Uno dei principali obiettivi futuri sarà quello di migliorare l'accuratezza nella classificazione della classe "Women", esplorando l'uso di altri modelli di rete o tecniche avanzate di manipolazione audio che possano preservare le caratteristiche rilevanti di questa classe. Questo miglioramento potrebbe aumentare l'efficacia complessiva dei modelli utilizzati. Una possibile strada sarebbe quella di adottare modelli

Audio Vision Transformer (Audio ViT). Questi modelli rappresentano una recente evoluzione delle reti neurali, basandosi sul meccanismo di self-attention dei Transformer, originariamente sviluppati per compiti di visione artificiale. Applicando questo concetto agli spettrogrammi audio, gli Audio ViT sono in grado di catturare con maggiore precisione le dipendenze locali e globali all'interno dei segnali audio. Il vantaggio principale rispetto alle reti neurali convoluzionali (CNN) risiede nella capacità dei Transformer di apprendere relazioni a lungo raggio all'interno dei dati. Mentre le CNN eccellono nell'individuare pattern locali, i modelli Transformer sono più efficaci nel considerare interazioni tra diverse porzioni di un segnale, anche quando queste sono distanti tra loro. Questo li rende particolarmente adatti ad analizzare dati audio complessi o soggetti a rumore. L'uso degli Audio ViT potrebbe risultare particolarmente utile per migliorare l'accuratezza nella classificazione della classe "Women", poiché questi modelli possono rilevare sottili caratteristiche distintive nei segnali vocali femminili. Trattando l'audio come una sequenza di frame, i Transformer permettono di mantenere intatte informazioni cruciali che potrebbero essere perse con tecniche di pre-processing o filtraggio più tradizionali. Questo approccio innovativo offre dunque un promettente miglioramento nella classificazione dei suoni, contribuendo a una maggiore precisione del modello nelle applicazioni audio.

Un'ulteriore direzione di sviluppo consiste nell'implementazione di modelli di rete neurale su dispositivi portatili, come smart band o applicazioni per smartphone. Tuttavia, un ostacolo significativo è rappresentato dall'uso continuo del microfono, essenziale per monitorare costantemente l'ambiente circostante. Questo utilizzo è spesso limitato dai sistemi operativi, principalmente per motivi legati alla privacy e alla durata della batteria. In particolare, sia Android che, in modo ancor più restrittivo, iOS impongono rigide restrizioni sull'accesso al microfono in background da parte delle applicazioni. Oltre a queste limitazioni software, è necessario considerare anche le restrizioni hardware dei dispositivi mobili, che possono influenzare le prestazioni dei modelli di rete neurale. Tuttavia, questo problema può essere parzialmente mitigato attraverso l'adozione di modelli ottimizzati per dispositivi mobili, come MobileNetV2. Questo modello è stato progettato per garantire un'alta efficienza computazionale e un ridotto consumo di risorse, mantenendo al contempo

prestazioni elevate. L'uso di modelli come MobileNetV2 consente l'implementazione di reti neurali su dispositivi portatili senza compromettere in modo significativo le prestazioni o l'autonomia del dispositivo. Di conseguenza, questa soluzione si presenta come promettente per applicazioni che richiedono monitoraggio continuo dell'audio, come quelle dedicate alla sicurezza o al riconoscimento di situazioni di pericolo.

Un aspetto innovativo da esplorare è la creazione di un sistema multimodale che integri l'analisi sia dell'audio che dei video provenienti da videocamere di sicurezza. Questo approccio consentirebbe di esaminare simultaneamente i segnali audio e i filmati, migliorando la capacità di rilevare potenziali minacce e identificare eventuali aggressori. Combinando le informazioni ottenute dai due tipi di dati, si potrebbe ottenere una visione più completa e accurata delle situazioni di pericolo. L'analisi multimodale presenta diversi vantaggi. In primo luogo, l'integrazione di dati audio e video permette di sfruttare le complementarità delle due modalità: mentre i segnali audio possono catturare suoni critici, come grida di aiuto o rumori sospetti, i filmati forniscono un contesto visivo che aiuta a identificare il comportamento degli individui coinvolti. Inoltre, l'uso di tecniche di machine learning avanzate, come le reti neurali convoluzionali (CNN) per l'analisi video e audio, può migliorare notevolmente le prestazioni del sistema. Un ulteriore aspetto da considerare è la necessità di una gestione efficiente dei dati, poiché l'elaborazione simultanea di flussi audio e video richiede risorse computazionali significative. L'adozione di architetture di rete neurale ottimizzate e strategie di compressione dei dati potrebbe aiutare a superare queste sfide. Inoltre, il sistema dovrebbe essere progettato per operare in tempo reale, garantendo una risposta rapida in situazioni di emergenza. Infine, l'integrazione di feedback e meccanismi di apprendimento continuo potrebbe consentire al sistema di migliorare le proprie capacità nel tempo, adattandosi a nuovi scenari e minacce. In questo modo, un sistema multimodale non solo aumenterebbe l'affidabilità del rilevamento delle situazioni di pericolo, ma contribuirebbe anche a una maggiore sicurezza.

In conclusione, l'adozione di tecniche avanzate, lo sviluppo di applicazioni su dispositivi mobili e l'implementazione di sistemi multimodali rappresentano le prossime tappe per migliorare l'efficacia del riconoscimento di situazioni di pericolo basato su reti neurali.

# Bibliografia


[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.

[3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[4] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[5] Sunil K Punjabi, Suvarna Chaure, Ujwala Ravale, and Deepti Reddy. Smart intelligent system for women and child security. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 451–454, 2018.

[6] G. Monisha, M. Monisha, Pavithra Gunasekaran, and Dr.Subhashini Radhakrishnan. Women safety device and application-femme. *Indian Journal of Science and Technology*, 9, 03 2016.

[7] Anand Jatti, Madhvi Kannan, R M Alisha, P Vijayalakshmi, and Shrestha Sinha. Design and development of an iot based wearable device for the safety and security of women and girl children. In *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, pages 1108–1112, 2016.

[8] Rabbina Ridan Khandoker, Shahreen Khondaker, Fatiha-Tus-Sazia, Fernaz Narin Nur, and Shaheena Sultana. Lifecraft: An android based application system for women safety. In *2019 International Conference on Sustainable*

*Technologies for Industry 4.0 (STI)*, pages 1–6, 2019.

[9] Awal Ahmed Fime, Md. Ashikuzzaman, and Abdul Aziz. Audio signal based danger detection using signal processing and deep learning. *Expert Systems with Applications*, 237:121646, 2024.

[10] Haziqa Sajid. Differenze tra machine learning e deep learning, 2023. https://www.unite.ai/it/differenze-chiave-tra-machine-learning-e-deep-learning/.

[11] Soumallya Bishayee. Artificial neural network, 2023. https://medium.com/@soumallya160/everything-you-need-to-know-about-artificial-neural-network.

[12] Mayank Mishra. Convolutional neural network, 2020. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

[13] Serkan Kızılırmak. Relu function, 2023. https://medium.com/@serkankizilirmak/rectified-linear-unit-relu-function-in-machine-learning-understanding-the-basics.

[14] Rafay Qayyum. Pooling layer, 2022. https://towardsai.net/p/l/introduction-to-pooling-layers-in-cnn.

[15] Adith Narein T. Inceptionv3 model. https://iq.opengenus.org/inception-v3-model-architecture/.

[16] Atul Pandey. Depth-wise convolution and depth-wise separable convolution, 2018. https://medium.com/@zurister/depth-wise-convolution-and-depth-wise-separable-convolution-37346565d4ec.

[17] Nitish Kundu. Exploring resnet50: An in-depth look at the model architecture and code implementation, 2023. https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f.

[18] Awais Ahmad, Muhammad Khan, Muhammad Javed, Majed Alhaisoni, Usman Tariq, Seifedine Kadry, Jungin Choi, and Yunyoung Nam. Human gait recognition using deep learning and improved ant colony optimization. *Computers, Materials and Continua*, 70:2261–2276, 02 2022.

[19] Gaudenz Boesch. Xception model: Analyzing depthwise separable convolutions, 2024. https://viso.ai/deep-learning/xception-model/.

[20] Ketan Doshi. Audio deep learning made simple (part 1): State-of-the-art techniques, towards data science, 2021. https://towardsdatascience.com/audio-deep-learning-made-simple-part-1-state-of-the-art-techniques-da1d3dff2504.

[21] Leland Roberts. Understanding the mel spectrogram, 2020. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

[22] Yusnita mohd ali, Paulraj M P, Sazali Yaacob, Raghad Yusuf, and Shahriman Abu bakar. Analysis of accent-sensitive words in multi-resolution mel-frequency cepstral coefficients for classification of accents in malaysian english. *International Journal of Automotive and Mechanical Engineering*, 7:1053–1073, 06 2013.